HIGH THROUGHPUT TERMINAL/CDMA MODEM FOR SATELLITE COMMUNICATIONS

FINAL REPORT

for

Rome Research Site

AFRL/IFGC
525 Brooks Road
Rome, New York 13441-4505

Contract: F30602-99-C-0110

SBIR TOPIC: AF99-132

January 14, 2000

Submitted By:

PREDICTION SYSTEMS, INC.
309 Morris Avenue
Spring Lake, NJ 07762
☎ (732)449-6800
🖷 (732)449-0897

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. AGENCY USE ONLY (*Leave Blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED

4. TITLE AND SUBTITLE HIGH THROUGHPUT TERMINAL/CDMA MODEM FOR SATELLITE COMMUNICATIONS	5. FUNDING NUMBERS F 30602-99-C-0110
6. AUTHOR(S) KENNETH IRVINE, WILLIAM C. CAVE	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) PREDICTION SYSTEMS, INC 309 MORRIS AVENUE - SUITE G SPRING LAKE, NJ 07762	8. PERFORMING ORGANIZATION REPORT NUMBER PSI-99002
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AIRFORCE RESEARCH LABORATORY INFORMATION DIRECTORATE 26 ELECTRONIC PARKWAY ROME, NY 13441-4514	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES NONE	
12a. DISTRIBUTION/AVAILABILITY STATEMENT UNLIMITED	12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

Communications on the battlefield can benefit from the use of satellites to support connectivity over widely dispersed deployments and rugged terrain. It appears that satellite communications for military purposes will be enhanced using Code Division Multiple Access (CDMA). The spread spectrum technology used for CDMA has provided terrestrial military communication systems with many positive features including anti-jamming, Low Probability of Detection (LPD), and Low Probability of Interception (LPI). CDMA can also be used to gain increased capacity, improved coverage, reduced transmission power, and extended range.

This proposal describes a CAD approach to rapid analysis and design of communications systems. Based upon PSI's real-time control and simulation environment, it has been used for development of dynamic real-time data and communication systems. This environment consists of the General Simulation System (GSS) for discrete event simulation and the Visual Software Environment (VSE) for building real-time control systems. On top of these tools, PSI has developed a new Run-Time Graphics (RTG) system that allows the analyst to graphically monitor and interact with the *live system* as well as a simulation. This environment is directly applicable to the rapid development of CDMA technology for use in satellite communications.

14. SUBJECT TERMS COMPUTER-AIDED DESIGN (CAD) CDMA MODELING & SIMULATION SATELLITE MODEM COMMUNICATIONS SYSTEMS	15. NUMBER OF PAGES 68
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNLIMITED

HIGH THROUGHPUT TERMINAL/CDMA MODEM FOR SATELLITE COMMUNICATIONS

FINAL REPORT

for

Rome Research Site

AFRL/IFGC
525 Brooks Road
Rome, New York 13441-4505

Contract: F30602-99-C-0110

SBIR TOPIC: AF99-132

January 14, 2000

Submitted By:

PREDICTION SYSTEMS, INC.
309 Morris Avenue
Spring Lake, NJ 07762
☎ (732)449-6800
🗎 (732)449-0897

TABLE OF CONTENTS

1. BACKGROUND

Communications on the battlefield can benefit from the use of satellites to support connectivity over potentially widely dispersed deployments and rugged terrain. It appears that satellite communications for military purposes will be enhanced using Code Division Multiple Access (CDMA).

CDMA is not a new technology. It has been used previously to develop terrestrial military communications systems. Until recently, CDMA technology has been difficult and expensive to implement in a cost effective manner. As a result, satellite communication systems have been traditionally built using Frequency Division Multiple Access (FDMA) and Time Division Multiple Access (TDMA) schemes. Recent development of high density digital integrated circuits, however, are now making CDMA technology economically feasible and equipment providers are now actively pursuing the insertion of CDMA into wireless communication systems including satellite systems.

The spread spectrum technology used for CDMA has provided terrestrial military communication systems with many positive features including anti-jamming, Low Probability of Detection (LPD), and Low Probability of Interception (LPI). CDMA can also be used to gain increased capacity, improved coverage, reduced transmission power, and extended range.

The Phase I approach to analysis and design of satellite communications using CDMA technology is based upon PSI's real-time control and simulation environment that has been used for rapid development of dynamic real-time data and communication systems. This environment consists of the General Simulation System (GSS) for discrete event simulation and the Visual Software Environment (VSE) for building real-time control systems. On top of these tools, PSI has developed a new Run-Time Graphics (RTG) system that allows the analyst to graphically monitor and interact with the live system as well as a simulation. This environment is directly applicable to the rapid development of CDMA technology for use in satellite communications.

2. INTRODUCTION

PSI's proposed approach is based upon computer technology that affords the implementation of powerful simulations on standard PCs and laptop computers. New laptops generally provide sufficient speeds of computation, main and auxiliary memory sizes, and graphics to accommodate state-of-the-art simulation and optimization.

Figure 2.1 is an illustration of the type of simulation envisioned. A simulation of the CDMA satellite system can be constructed using previously developed models represented by icons. These models can be parameterized and thus tailored to the specific scenarios to be represented. The analyst can construct a scenario by interconnecting icons representing previously developed models of communication assets including satellite modems. The simulation can then be used to assess the performance of the system to various CDMA parameter changes. This affords an analyst the ability to quickly determine how the CDMA satellite data terminal design can be improved, particularly to support mission critical tasks.

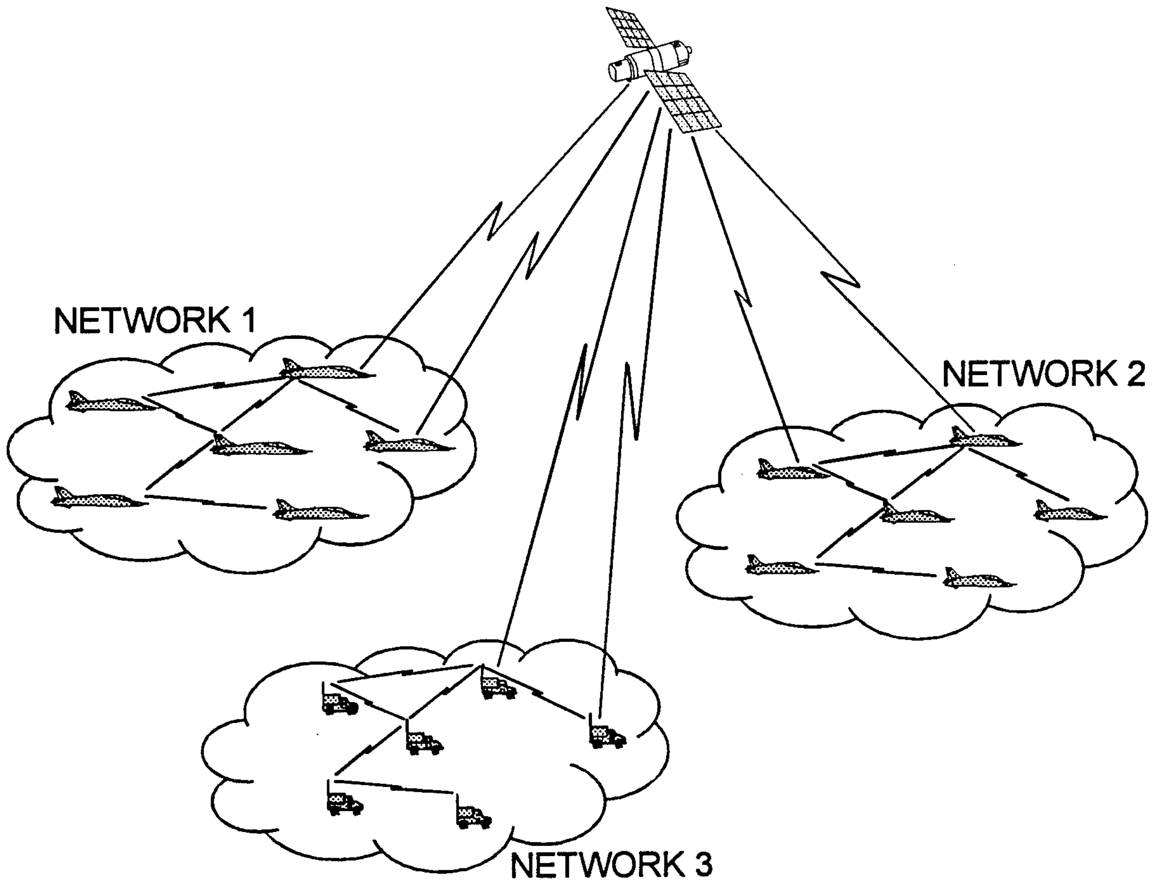

Figure 2.1. CDMA Satellite System Equipped Tactical Networks.

3. REVIEW OF WORK DONE

This report covers the work done on the SBIR Phase I contract, AF Topic No. 99-132, “High Throughput Terminal/CDMA Model for Satellite Communications.” During this period, PSI has completed implementation and test of a Phase I version of the CDMA Satellite simulation and analysis tool. Delivery of this working tool to Rome completes 100% of the proposed Phase I work effort. As part of this delivery, PSI provided AFRL/IFGC with an upgraded version of GSS Release 8.0, the simulation environment, and scenarios and test demonstrations to the Air Force for use at AFRL, Rome.

In prior reporting periods, PSI produced the design and implementation of the models proposed for use in the CDMA Satellite simulation and analysis tool. In this final report we will describe the models that have been implemented, tested, and prepared for delivery to AFRL. PSI will deliver the simulation conduct demonstrations on 18 JAN 00 at AFRL/IFGC, Rome, NY.

3.1 OVERVIEW OF THE SIMULATION

The Phase I CDMA satellite simulation, SAT_COMM, that PSI will deliver is shown in Figure 3.1. A full set of engineering drawings and documentation for the simulation are provided with the delivery package. This simulation contains a number of a modified models from an existing PSI simulation that contains the SINCGARS and HAVE-QUICK radios. This simulation was selected because it contains most of the models required to support the proposed Phase II design effort. It also includes models of both TCP and UDP protocol stacks, a MIL-STD-188-220 stack that contains an IP layer model, an intranet layer model, the MIL-220 data link layer model, and a MIL-220 physical layer model.

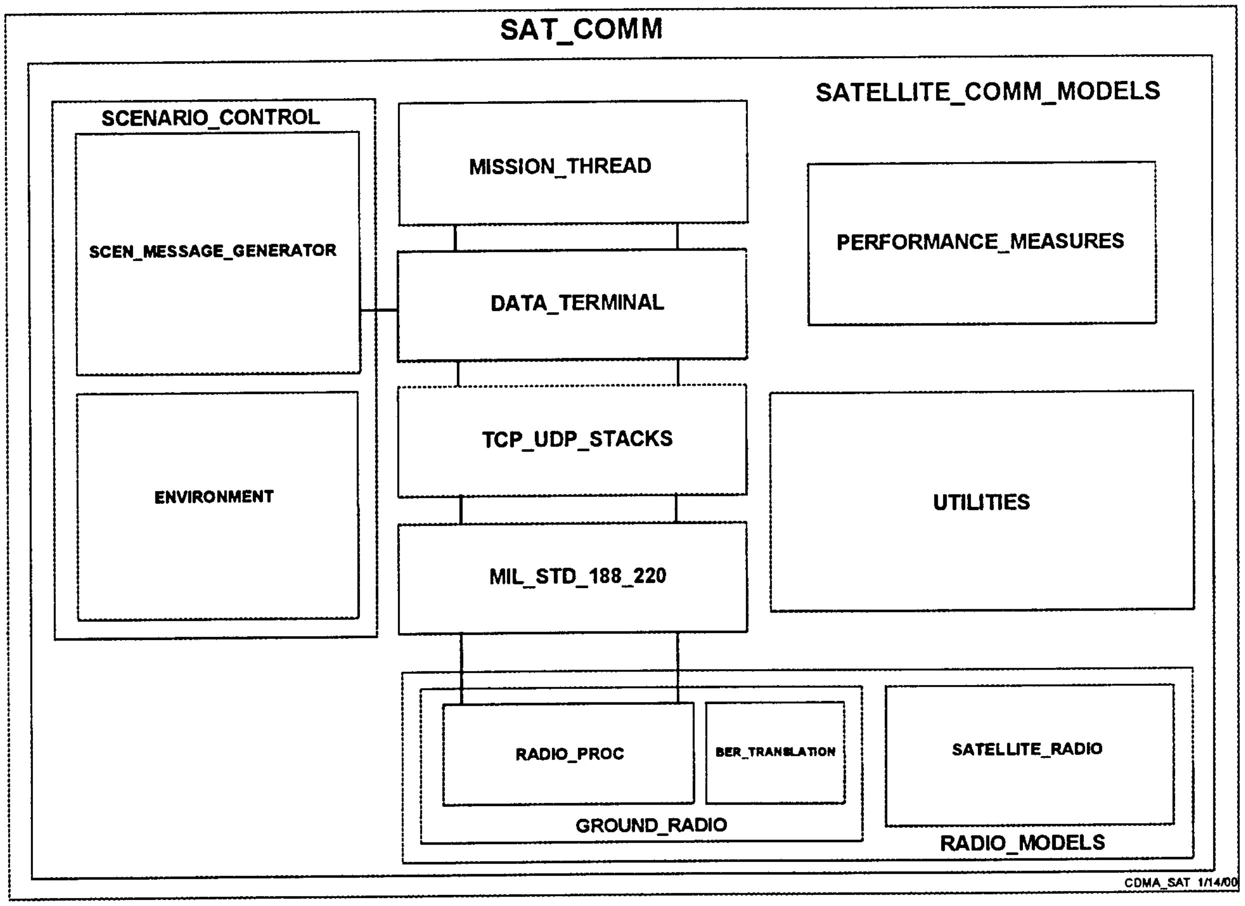

Figure 3.1. Overview of SAT_COMM simulation delivered under Phase I.

Although not shown, this simulation's ENVIRONMENT model contains the Fast Propagation Prediction System (FPPS) developed for both Army and Joint agencies. FPPS provides fast accurate propagation path loss calculations for frequencies from 20 MHz to 20 GHz using modified versions of ECAC's TIREM. FPPS has been validated using data from many field tests and compared with TIREM. It is much faster, and has been shown to be more accurate in most of these comparisons.

Supporting FPPS is a Selected Area Terrain (SAT) data management system. This system takes in the National Imagery and Mapping Agency (NIMA) CD-ROMs that contain the U.S. Government's Digitized Terrain Elevation Data (DTED) in WGS-84 format and creates an SAT database for the playbox of interest. The SAT database eliminates nonlinear equations when the playbox covers multiple DTED grid zones, making the calculations very fast.

The ability to calculate propagation path loss due to terrain and foliage allows for evaluation of satellite relays to extend communications between units located on the ground as shown in Figure 3.2. Measures of performance can compare the number of hops and time it takes to pass messages or files using satellite gateways versus ground gateways.

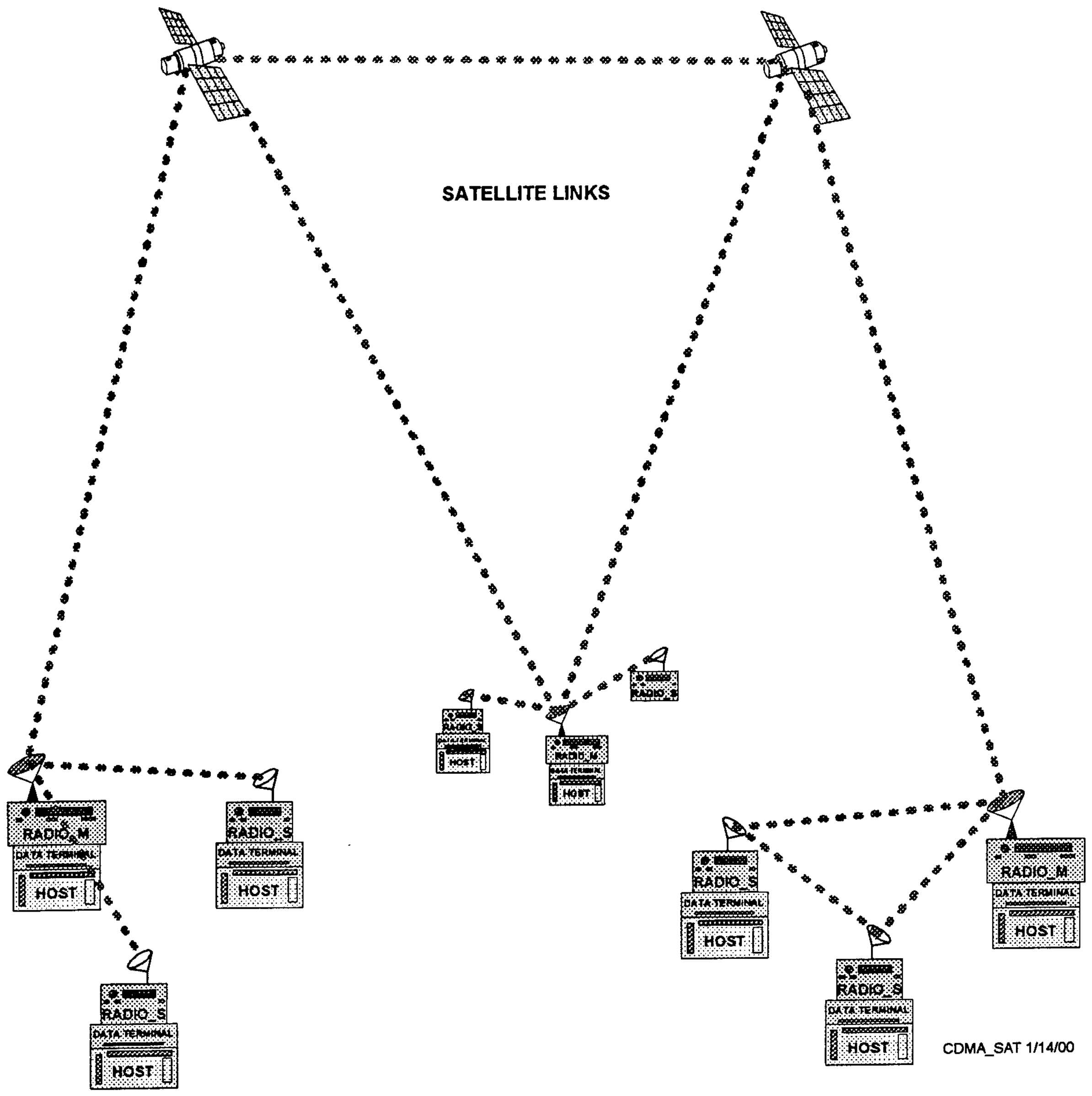

Figure 3.2. SAT_COMM provides for gateways with data terminals in different nets.

The SAT data management system contains facilities for generating contour draw files from the same SAT database that is used for computing the propagation path loss. The model also contains the draw programs for displaying the terrain contours as a background overlay behind the dynamic simulation graphics. This provides precise correlation between the positioning of an icon on the screen with respect to the terrain contours, and the expected variations in path loss between two radios.

This simulation contains the facilities for taking in needline files and mission threads, deployments, and radio characteristics, as well as characteristic data for all of the protocol layers. It also contains various measures of performance that are appropriate for the satellite simulation.

3.2 IMPLEMENTATION OF THE PHASE I SIMULATION

This simulation will be delivered prior to the end of the Phase I contract. It will provide an interactive graphical interface. This interface will allow users to create a network of ground radios that communicate via the satellite. Users will be able to deploy one or more satellites and ground radios that are designated to be on specified satellite networks.

Gateways between networks can be specified by simply connecting radios of different networks. Connectivity between units is automatically determined using the FPPS propagation system and DMA terrain data. Bit error ratios for individual links can be overridden and directly specified to support the analysis of scenarios requiring specific connectivity. All graphical interactions are recorded in output files which can later be used as input files to reproduce the order of deployment updates. Deployment output and input files have the same format for easily reusing output files as input files. Scenarios can be run entirely from input files if desired.

Network performance is shown on the graphics workstation while the simulation is running. The styles of the lines between units indicates the Bit Error Ratio (BER) of that link. The more solid the line is, the better the connectivity is. The color of the link is used as an indication of its activity. Graphical instruments are also used to indicate overall scenario performance measures that are continuously updated as the simulation is running.

Message traffic can be generated in 2 ways. Individual messages can be specified and sent at a given time from a message script file. Alternatively, a needline between a given source and destinations can be specified to stochastically generate messages. The needline indicates when the first message should be generated, how many messages should be generated, and the mean generation time between messages. Either way of generating messages supports specifying the message size, its priority, and up to 16 destinations. To support mission analyses, the simulation supports the use of mission threads. The mission thread indicates the procedure of each type of unit when it receives a particular message type within a mission thread.

The MIL-STD-188-220A layers are modeled in detail. The intranet layer supports dynamic routing decision making as a result of changing connectivity. The data link layer supports the various data link types and their operating parameters including forward error correction, time dispersal coding, and concatenation. The physical layer includes support for interfacing to a SINCGARS or HAVE QUICK radio including the robust communication protocol.

The simulation supports the use of TCP, UDP, IP, and Segmentation and Reassembly protocol layers. These layers are supported by individual models for each and reside above the MIL-STD-188-220A layers.

All protocol layer models in the simulation support the specification of their operational parameters. These parameters can be used to modify the behavior of the protocol to test a protocol layer's effect on network performance under various scenarios. The performance measures model is used to track the performance of the network and to provide the necessary statistics at the conclusion of the simulation.

The SAT_COMM simulation was designed along the ISO 7-layer protocol framework. The simulation consists of the following models:

- SCENARIO_CONTROL Model
- PERFORMANCE_MEASURES Model
- MISSION_THREAD Model
- DATA_TERMINAL Model
- COUL Model
- TCP Model
- IP Model
- MIL_220_INTRANET Model
- MIL_220_DATA_LINK_LAYER Model
- MIL_220_PHYSICAL_LAYER Model
- SINCGARS/HAVE_QUICK Model
- SATELLITE Model
- GRAPHICS_UTILITY Model

These models are described in detail in Section 4. Table 3.1 shows how the protocol layer models map to the 7-layer ISO standard.

All models were developed using the General Simulation System (GSS). GSS provides a complete simulation environment for developing large-scale complex systems. This system supports the design, development, and support of simulations using engineering drawings to graphically illustrate the simulation architecture.

GSS relies on 2 fundamental paradigms for developing large scale simulation systems. These are the separation of data (Resources) from instructions (Processes) and the separation of architecture from language. This separation allows the simulation architecture to be developed before any code is written. The development of the architecture results in engineering drawings that provide a one-to-one mapping, without abstractions, to the language elements (Resources and Processes). The implementations of the resources and processes are performed in the language environment.

The SAT_COMM simulation has 4 engineering drawings. The first shows the overall simulation consisting of the SAT_COMM model, the TCP_UDP_STACKS models, and the MIL_STD_188_220 model. The SAT_COMM model includes the details of all the models except for the protocol stacks. The protocol stacks are mapped into the ISO framework in Table 3.1. The reader is referred to the engineering drawings of the three models shown in Figure 3.2

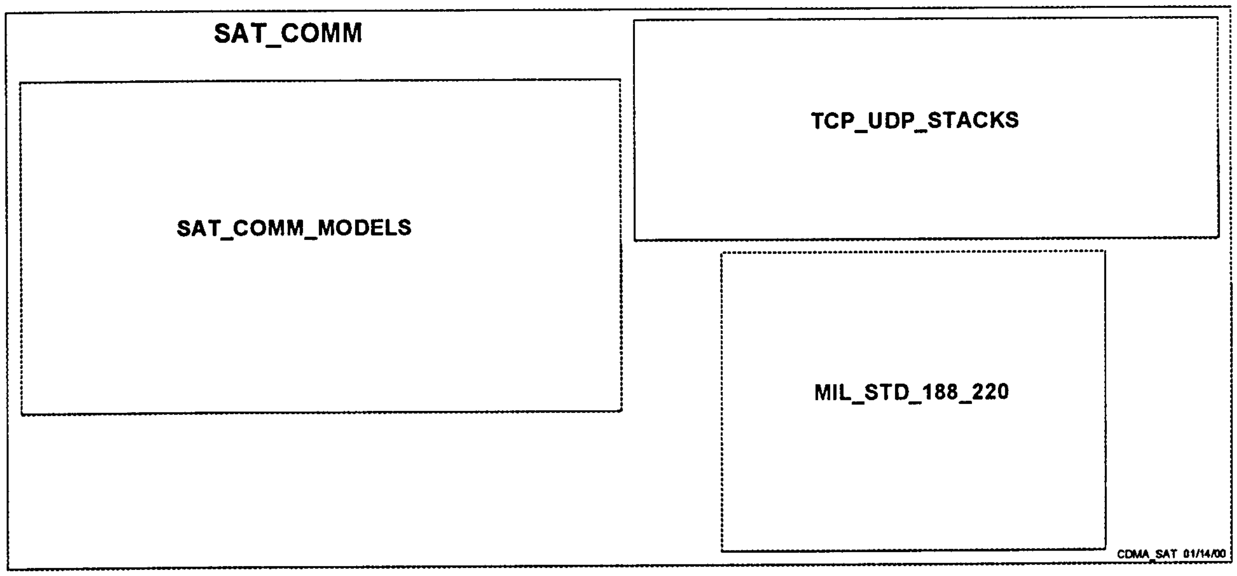

Figure 3.2. SAT_COMM provides for gateways with data terminals in different nets.

GSS supports the ability to easily export and import models out of and into GSS directories. This allows various implementations of a model to be quickly swapped into a simulation directory for testing. For the protocol analysis simulations, this supports the ability to swap in different implementations of TCP/UDP protocol layers.

Table 3.1 GSS Model to ISO Layer Mapping

LAYER	DESCRIPTION	GSS MODEL
7	APPLICATION	
6	PRESENTATION	COUL
5	SESSION	
4	TRANSPORT	TCP
3B	INTERNET	IP
3A	INTRANET	MIL_220_INTRANET
2	DATA LINK	MIL_220_DATA_LINK_LAYER
1	PHYSICAL	MIL_220_PHYSICAL_LAYER

To make the model implementations independent of each other, each protocol layer model reads the parameters that it needs from its own files. Therefore, when a new model is included within the simulation, the modifications to support any new parameters will arrive with the model. Parameter input files and output files containing statistics were developed using the Standard File Interface (SFI) format. This format was developed to provide a well defined consistent means of supporting input and output files.

3.3 PROPOSED REVISIONS FOR THE PHASE II SIMULATION

The simulation delivered for the Phase I contract follows the architecture of the prior simulation from which it was derived. Although this simulation provides the facility to interact with it while it is running, including the ability to insert, move, and delete units, the old architecture makes it difficult to add more models. The new architectural approach used by PSI in the Defensive Information Operations Planning Tool (DIOPT) simulations allow one to add different radios and ground systems (either wired or wireless) as well as satellites.

Based upon the DIOPT architecture, PSI has proposed using the DIOPT archicture to add models. This implies taking the models from this simulation and incorporating them into the new architecture. This will make it much easier to provide a common graphical interface to all of the models, as well as standard interfaces between equipment and link models.

4. MODEL DESCRIPTIONS

This section describes the models used to develop the SAT_COMM simulation. The reader is referred to the engineering drawings of the three models shown in Figure 3.2

4.1 SAT_COMM MODELS

4.1.1 SCENARIO_CONTROL MODEL

The SCENARIO_CONTROL Model is used to provide input deployment and traffic data to the simulation as well as control the parameters used when performing multiple simulation runs. It is a hierarchical model consisting of the following submodels that are described in the following subsections:

- SCEN_MESSAGE_GENERATOR
- ENVIRONMENT

4.1.1.1 SCEN_MESSAGE_GENERATOR MODEL

The SCEN_MESSAGE_GENERATOR model performs the following tasks:

- Read in needline and scripted data from files.
- Initiate the transfer of messages to the DATA_TERMINAL Model.

The SCENARIO_CONTROL Model was developed to support both needline and scripted input. The message and needline input SFI files are time ordered and can both be used simultaneously to generate messages within the system. Each message generated may be an individual message to a set of destinations or may be the first message of a mission thread to be started. Each message generated is given a unique ID within the data controller model to facilitate tracking the message within the system. The messages are designed to support up to 16 destinations.

The needline and scripted input methods are described below. Figures 4.1.1.1 and 4.1.1.2 show example Needline and Message Script files.

NEEDLINE INPUT

The needline input file is used to represent a user by stochastically generating messages. The needline input file contains the following information:

- **Needline ID** - Needline Identification used to correlate the output with the input.
- **Start Time (seconds)** - Time to enter the message into the system.
- **Mission Thread ID** - Identification of a mission thread from the mission thread SFI input file. A value of 0 indicates that this is a point-to-point needline that is not part of a mission thread. If a valid mission thread id is provided, all fields after the source unit ID are ignored except for Max Messages. The Max Messages field indicates how many missions should be started.
- **Source Unit ID (NNHHH)** - Network/Host pair identifying the source.
- **Destination Unit IDs (NNHHH)** - Up to 16 Network/Host pairs identifying the destinations. Destinations are read until a 0 ID is encountered.
- **Precedence** - Message precedence between 1 (lowest) and 5 (highest). These precedence levels are mapped into other precedence levels at lower layers. There may not be a one-to-one mapping of these precedence levels to lower layer precedence levels.
- **Message Length (octets)** - Size of the message to be transmitted. The maximum message size allowed is based on the maximum segment size specified in the segmentation and reassembly SFI input file. The segmentation and reassembly model does not support messages that exceed 256 segments.
- **Mean Generation Time (seconds)** - Average time between generating messages on the needline.
- **Max Messages** - Number of messages to generate on the needline before the needline ends.

```
*** NEEDLINE.DAT INPUT FILE
***
*NLID STRTTIME MTH SRCE DST1 DST2 DST3 DST4 DST5 DST6 DST7 DST8 DST9 DSTA DSTB DSTC DSTD DSTE DSTF DSTG PR LENGTH GEN-TIME MXMSG
00001 00010.00 000 5003 5004 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 03 001000 00020.00 00010
00002 00030.00 001 4005 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 03 001000 00020.00 00003
```

Figure 4.1.1.1 Needline Input File (NEEDLINE.DAT)

SCRIPTED INPUT

The scripted input file is used to generate messages at a given time with a specified size. The scripted input file contains the following information:

- **Start Time (seconds)** - Time to enter the message into the system
- **Message ID** - Message Identification used to correlate the output with the input.
- **Mission Thread ID** - Identification of a mission thread from the mission thread SFI input file. A value of 0 indicates that this is a point-to-point needline that is not part of a mission thread. If a valid mission thread id is provided, all fields after the source unit ID are ignored except for Max Messages. The Max Messages field indicates how many missions should be started.
- **Source Unit ID (NNHHH)** - Network/Host pair identifying the source.
- **Destination Unit IDs (NNHHH)** - Up to 16 Network/Host pairs identifying the destinations. Destinations are read until a 0 ID is encountered.
- **Precedence** - Message precedence between 1 (lowest) and 5 (highest). These precedence levels are mapped into other precedence levels at lower layers. There may not be a one-to-one mapping of these precedence levels to lower layer precedence levels.
- **Message Length (octets)** - Size of the message to be transmitted. The maximum message size allowed is based on the maximum segment size specified in the segmentation and reassembly SFI input file. The segmentation and reassembly model does not support messages that exceed 256 segments.

```
*** MESSAGE.DAT SCRIPT INPUT FILE
***
*SIMTIME MSGID MTH SRCE DST1 DST2 DST3 DST4 DST5 DST6 DST7 DST8 DST9 DSTA DSTB DSTC DSTD DSTE DSTF DSTG PR
LENGTH
00500.00 00001 001 4005 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 00
000000
00550.00 00002 000 1003 1002 3003 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 01
000500
00600.00 00004 001 4005 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 00
000000
```

Figure 4.1.1.2 Scripted Input File (MESSAGE.DAT)

4.1.1.2 ENVIRONMENT MODEL

The environment model is used to read in the deployment and connectivity files and compute the BER values for each link in the deployment. The simulation supports 2 methods of building the connectivity table and both can be used together concurrently.

The deployment file is a time ordered file of unit events that define how the deployment is modified in time. Figure 4.1.2.1 shows a sample deployment input file. Typically, the initial deployment is defined at the beginning of the simulation and unit updates occur later in time. The deployment is used to compute the BER between all units within a network by calculating a Signal-to-Noise Ratio (SNR) and translating this ratio to the BER by using the radio characteristic curve. The Fast Propagation Prediction System library (FPPS.a) is used to compute free space and excess terrain loss. The deployment file consists of set of records containing the following information:

- **Sample Number (seconds)** - This field is contained in all SFI files and is not a user formatted field. It contains the time that the unit should be added to the simulation or updated if it was already added.
- **IP Address (NNHHH)** - IP address of the unit to be added or updated. The first 2 digits are the units network address and the last 3 digits are its host address within the network.
- **Zone/Cell (ZZZCC)** - Zone and cell id for the units MGR coordinates.
- **X Coordinate (meters)** - MGR X coordinate.
- **Y Coordinate (meters)** - MGR Y coordinate.
- **Power (watts)** - Units transmitting power in watts.
- **Antenna Height (meters)** - Height of the units antenna with respect to the ground.
- **Antenna Gain (dB)** - Units antenna gain.
- **Cable Loss (dB)** - Units cable loss.

```
*
*  DEPLOYMENT INPUT SFI FILE
*
*  CASESAMPLESEQCODE1111111111112222222222222333333333333444444444444555555555555666666666666777777777777888888888888
H     1 8S             IP ADDRESS  ZONE/CELL    X COORD      Y COORD       POWER     ANT HEIGHT   ANT GAIN    CABLE LOSS
F                    I       NNHHHA           I            I            F2  WATTS    F1  METERS  F1    DB       F1    DB
              0              0100132UNB                05000        30000        4.50         3.0          6.0          1.5
              0              0100232UNB                10000        30000        4.50         3.0          6.0          1.5
              0              0100332UNB                10000        35000        4.50         3.0          6.0          1.5
              0              0100432UNB                10000        20234        4.50         3.0          6.0          1.5
              0              0100532UNB                17929        26211        4.50         3.0          6.0          1.5
              0              0200132UNB                27968        22226        4.50         3.0          6.0          1.5
              0              0200232UNB                35976        24219        4.50         3.0          6.0          1.5
              0              0200332UNB                37000        33000        4.50         3.0          6.0          1.5
              0              0200432UNB                28000        31000        4.50         3.0          6.0          1.5
              0              0200532UNB                18000        26000        4.50         3.0          6.0          1.5
              0              0300132UNB                20000        17226        4.50         3.0          6.0          1.5
              0              0300232UNB                10000        15000        4.50         3.0          6.0          1.5
              0              0300332UNB                08000        08000        4.50         3.0          6.0          1.5
              0              0300432UNB                10000        20000        4.50         3.0          6.0          1.5
              0              0300532UNB                03000        17000        4.50         3.0          6.0          1.5
              0              0400132UNB                36992        15742        4.50         3.0          6.0          1.5
              0              0400232UNB                20000        17000        4.50         3.0          6.0          1.5
              0              0400332UNB                28000        22000        4.50         3.0          6.0          1.5
              0              0400432UNB                26000        12000        4.50         3.0          6.0          1.5
              0              0400532UNB                28000        07000        4.50         3.0          6.0          1.5
              0              0500132UNB                37000        16000        4.50         3.0          6.0          1.5
              0              0500232UNB                36000        24000        4.50         3.0          6.0          1.5
              0              0500332UNB                45000        30000        4.50         3.0          6.0          1.5
              0              0500432UNB                45000        20000        4.50         3.0          6.0          1.5
              0              0500532UNB                47000        17000        4.50         3.0          6.0          1.5
```

Figure 4.1.2.1 Deployment SFI Input File (DEPLOY.SFI)

In addition to the deployment file, the connectivity input file can be used to explicitly override the computed BER values for specified links. Figure 4.1.2.2 shows a sample connectivity input file. As with the deployment file, this file is also time ordered. The file has the following fields:

- **Address 1 (NNHHH)** - Network/Host ID of the link source.
- **Address 2 (NNHHH)** - Network/Host ID of the link destination.
- **BER** - BER of the link.

```
*
*  CONN INPUT SFI FILE
*
*  CASESAMPLESEQCODE111111111111222222222222333333333333
H     1 3S              IPADDR 1    IPADDR 2      BER
F                   I           I           E
                           00000       00000       .1E+01
                           01001       01002       .0E+01
                           01002       01003       .0E+01
                           01002       01004       .0E+01
                           01004       01005       .0E+01
                           02001       02002       .0E+01
                           02002       02003       .0E+01
                           02002       02004       .0E+01
                           02004       02005       .0E+01
                           03001       03002       .0E+01
                           03002       03003       .0E+01
                           03002       03004       .0E+01
                           03004       03005       .0E+01
                           04001       04002       .0E+01
                           04002       04003       .0E+01
                           04002       04004       .0E+01
                           04004       04005       .0E+01
                           05001       05002       .0E+01
                           05002       05003       .0E+01
                           05002       05004       .0E+01
                           05004       05005       .0E+01
```

Figure 4.1.2.2 Connectivity SFI Input File (CONN.SFI)

If both the source and destination of the link are 0 for the first record, the simulation will set all defined links in the simulation to the corresponding BER value provided. All other records are used to define the connectivity for specific links.

The BERs in the simulation will contain the values last computed by the deployment update or the connectivity file whichever is most recent. For deployment updates and connectivity updates that occur at the same time, the deployment update is performed first.

The simulation supports 3 types of radio networks. The network type SFI input file will be used to indicate the type of radio and the maximum transmission unit size being used for each network. Table 4.1.1 shows the radio network types that are supported. A network ID of 0 will be used to specify the default radio type to use for any unspecified networks. If no default is specified, the default will be assigned to the SINCGARS ICOM radio with a maximum transmission unit of 576 octets. Figure 4.1.2.3 shows a sample network type SFI input file.

Table 4.1.1 Network Type ID Assignments

Radio Used	Network Type
SINCGARS ICOM	ICOM
SINCGARS SIP	SIP
HAVE QUICK II	HQ

```
*
*  NETWORK_TYPE INPUT SFI FILE
*
*  CASESAMPLESEQCODE1111111111122222222222223333333333333
H     1 3S              NETWORK      NET TYPE      NET MTU
F                     I           A            I  OCTETS
                                 0ICOM                576   DEFAULT SETTING
                                 1ICOM                576
                                 2ICOM               1024
                                 3ICOM               1024
                                 4SIP                 576
                                 5HQ                  576
```

Figure 4.1.2.3 Network Type SFI Input File (NET_TYPE.SFI)

To support the computation of the link SNR values, an environment file is read in at the beginning of the simulation to specify the additional parameters used in the computation. Figure 4.1.2.4 shows a sample environment input file. The file consists of the following parameters:

- **Environment Noise (dB)** - Noise to use in computing the SNR.
- **Maximum Link BER** - Threshold of the maximum BER before a link is considered unusable.
- **ICOM Frequency (MHz)** - Frequency that the SINCGARS ICOM radios operate at. An average frequency in the middle of the radios frequency range is sufficient.
- **SIP Frequency (MHz)** - Frequency that the SINCGARS SIP radios operate at. An average frequency in the middle of the radios frequency range is sufficient.
- **HQ Frequency (MHz)** - Frequency that the Have Quick radios operate at. An average frequency in the middle of the radios frequency range is sufficient.

```
*
*  ENVIRONMENT PARAMETER SFI FILE
*
*  CASESAMPLESEQCODE11111111111122222222222223333333333333
H     1 3S            ENVIRONMENT PARAMETERS     VALUE
F                   A            A          F5
                    ENVIRONMENT NOISE         -115.00000  DB
                    MAX_BER_THRESHOLD            0.10000
                    ICOM FREQUENCY              50.00000  MHZ
                    SIP  FREQUENCY              50.00000  MHZ
                    HQ   FREQUENCY             300.00000  MHZ
```

Figure 4.1.2.4 Environment SFI Input File (ENVIRON.SFI)

For units that are deployed from the graphics, certain default parameters are assigned to them. These parameters are specified in the default input SFI file. Figure 4.1.2.5 shows a sample default input SFI file. The parameters provided include:

- **Default Power (Watts)** - Radio transmission power.
- **Default Antenna Height (Meters)** - Radio antenna height from ground.
- **Default Antenna Gain (dB)** - Radio antenna gain.
- **Default Cable Loss (dB)** - Cable loss of radio.

```
*
*  DEFAULT DEPLOYMENT PARAMETERS SFI FILE
*
*  CASESAMPLESEQCODE11111111111122222222222223333333333333
H     1 3S            DEFAULT PARAMETERS        VALUE
F                   A            A          F5
                    DEFAULT POWER                4.50000  WATTS
                    DEFAULT ANTENNA HEIGHT       3.00000  METERS
                    DEFAULT ANTENNA GAIN         6.00000  DB
                    DEFAULT CABLE LOSS           1.50000  DB
```

Figure 4.1.2.5 Default SFI Input File (DEFAULT.SFI)

While the simulation is running, a deployment output file is being generated. This file has the same format as the input file and contains all the events read in from the input file as well as any events that were entered via the graphics workstation. Figure 4.1.2.6 shows a sample deployment output file.

```
*
* DEPLOYMENT OUTPUT SFI FILE
*
H     1 8              IP ADDRESS  ZONE/CELL    X COORD     Y COORD       POWER     ANT HEIGHT   ANT GAIN   CABLE LOSS
F                    I       NNHHHA           I           I           F2  WATTS   F1  METERS  F1    DB      F1   DB
            0              100132UNB               5000       30000          4.50        3.0         6.0         1.5
            0              100232UNB              10000       30000          4.50        3.0         6.0         1.5
            0              100332UNB              10000       35000          4.50        3.0         6.0         1.5
            0              100432UNB              10000       20234          4.50        3.0         6.0         1.5
            0              100532UNB              17929       26211          4.50        3.0         6.0         1.5
            0              200132UNB              27968       22226          4.50        3.0         6.0         1.5
            0              200232UNB              35976       24219          4.50        3.0         6.0         1.5
            0              200332UNB              37000       33000          4.50        3.0         6.0         1.5
            0              200432UNB              28000       31000          4.50        3.0         6.0         1.5
            0              200532UNB              18000       26000          4.50        3.0         6.0         1.5
            0              300132UNB              20000       17226          4.50        3.0         6.0         1.5
            0              300232UNB              10000       15000          4.50        3.0         6.0         1.5
            0              300332UNB               8000        8000          4.50        3.0         6.0         1.5
            0              300432UNB              10000       20000          4.50        3.0         6.0         1.5
            0              300532UNB               3000       17000          4.50        3.0         6.0         1.5
            0              400132UNB              36992       15742          4.50        3.0         6.0         1.5
            0              400232UNB              20000       17000          4.50        3.0         6.0         1.5
            0              400332UNB              28000       22000          4.50        3.0         6.0         1.5
            0              400432UNB              26000       12000          4.50        3.0         6.0         1.5
            0              400532UNB              28000        7000          4.50        3.0         6.0         1.5
            0              500132UNB              37000       16000          4.50        3.0         6.0         1.5
            0              500232UNB              36000       24000          4.50        3.0         6.0         1.5
            0              500332UNB              45000       30000          4.50        3.0         6.0         1.5
            0              500432UNB              45000       20000          4.50        3.0         6.0         1.5
            0              500532UNB              47000       17000          4.50        3.0         6.0         1.5
```

Figure 4.1.2.6 Deployment SFI Output File (DEP_OUT.SFI)

4.1.2 PERFORMANCE_MEASURES MODEL

The PERFORMANCE_MEASURES Model is used to keep track of message statistics to be used in generating the performance measures needed to compare the protocols. The statistics are reported on in 3 formats as follows:

- Message Statistics
- Mission Thread Statistics
- Composite Statistics

Each of these is described in further detail in the following subsections:

MESSAGE STATISTICS

The simulation can track up to 1,000 messages with up to 16 destinations per message. Figure 4.2.1 shows a sample message statistics output file. Statistics are also tracked for messages that are generated to support mission thread links. The message statistics that are reported for each message include the following information:

- **Internal Message Id** - used by the simulation to track the messages.
- **Needline Number** - Number of the needline read in or 0 if the message was generated from the message script.
- **Message Id** - Id of the message read in from the script or the incremental message number if generated by a needline.
- **Thread Link** - Id of the link within the thread. Links are sequentially numbered within the thread in the simulation.
- **Destination Id** - Id of the message destination. Each destination for a message has its own entry to track the statistics of the delivery of the message to the given destination.
- **Transmit Start Time** - Simulation time that the speed-of-service timer starts. This is when the message is introduced to the application layer from the data controller.
- **Receive Time** - Simulation time that the speed-of-service timer stops. This is when the message is correctly received at the data controller from the application layer.
- **Message Delay** - Amount of time elapsed from the transmit start time to the receive time. A message delay of 0 indicates that the message was not successfully received.

```
*
*  MESSAGE STATISTICS OUTPUT FILE
*
H    1 8                 ID      NL      MSG   THRD LINK     DEST  START TIME    RECEIVE    MSG DELAY
F     S        I          I       I       I            I       F3  SECONDS F3  SECONDS F3   SECONDS
     1     0              1       1       1            0     5004     510.000     573.022      63.022
     1     0              2       0       1            1     4004     510.000     511.803       1.803
     1     0              3       0       1            2     2003     521.803     549.254      27.451
     1     0              4       1       2            0     5004     532.866     584.127      51.261
     1     0              5       2       1            1     4004     540.000     540.534        .534
     1     0              6       1       3            0     5004     546.172     596.869      50.696
     1     0              7       0       2            0     1002     550.000     568.827      18.827
     1     0              7       0       2            0     3003     550.000     605.860      55.860
     1     0              8       2       1            2     2003     550.534     560.159       9.625
     1     0              9       2       2            1     4004     552.025     552.559        .534
     1     0             10       0       1            3     2004     559.254     594.429      35.175
     1     0             11       2       3            1     4004     562.025     562.559        .534
     1     0             12       2       2            2     2003     562.559     651.575      89.016
     1     0             13       2       1            3     2004     570.159     599.107      28.948
     1     0             14       2       3            2     2003     572.559     651.576      79.017
                                                 .                                .
               .                                 .                                .
               .                                 .                                .
               .
```

Figure 4.2.1 Message Statistics SFI Input File (MSG_STAT.SFI)

MISSION THREAD STATISTICS

Figure 4.2.2 shows a sample mission threads statistics output file. The mission thread statistics include the following:

- **Internal Mission Thread Id** - used by the simulation to track invoked mission threads.
- **Mission Thread Id** - Mission thread identifier.
- **Needline Number** - Number of the needline read in or 0 if the message was generated from the message script.
- **Message Id** - Id of the message read in from the script or the incremental message number if generated by a needline.
- **Mission Thread Start Time (seconds)** - Simulation time that the mission starts. This is the time that the first message is generated on the mission thread.
- **Mission Thread Stop Time (seconds)** - Simulation time that the mission stops. This is the time that the last message of the mission is received at the destination.
- **Mission Delay (seconds)** - Amount of time needed to complete the mission. This is the difference between the mission thread stop and start times. A delay of 0 indicates that the mission was not completed.
- **Percent of Satisfied Links (%)** - Percentage of mission thread links that were successful for that mission thread.
- **Average Link Delay (seconds)** - Average delay of successful links of the mission thread.

```
*
*  MESSAGE THREAD STATISTICS OUTPUT FILE
*
H     1 9                      ID     MISSION          NL      MESSAGE     START        STOP         DELAY    % LINKS SAT   AVG DELAY
F      S           I            I           I           I           F3  SECONDS F3   SECONDS F3   SECONDS F3        %    F3   SECONDS
      1      0                  1           1           0            1   500.000     1272.059      772.059      100.000        42.243
      1      0                  2           1           2            1   530.000     1299.099      769.099      100.000        41.880
      1      0                  3           1           2            2   542.025     1358.218      816.194      100.000        41.016
      1      0                  4           1           2            3   550.420     1404.504      854.083      100.000        47.790
      1      0                  5           1           0            4   600.000     1445.946      845.946      100.000        41.423
```

Figure 4.2.2 Mission Thread Statistics SFI Input File (THRDSTAT.SFI)

COMPOSITE STATISTICS

Figure 4.2.3 shows a sample composite statistics output file. The composite statistics that are reported on include the following information (message statistics are for all messages regardless of precedence):

- **Total Number of Messages** - Total number of messages that were generated.
- **Number of Successful Messages** - Total number of messages that were successfully received. These messages are used to determine the average statistics.
- **Average Message Delay** - Average amount of time elapsed from first transmission attempt to correct reception of message.
- **Total Number of Mission Threads** - Total number of mission threads that were invoked.
- **Number of Successful Mission Threads** - Total number of mission threads where all the mission thread links were successful.
- **Average Thread Delay (seconds)** - Average delay for all successful threads.

For each precedence level, the following statistics are provided:

- **Total Number of Messages** - Total number of messages that were generated for that precedence.
- **Number of Successful Messages** - Total number of messages that were successfully received for that precedence. These messages are used to determine the average statistics.
- **Average Delay** - Average amount of time elapsed from first transmission attempt to correct reception of message for that precedence.

```
*
*  COMPOSITE STATISTICS OUTPUT FILE
*
H     1 3              COMMENT                         VALUE
F      S               A           A           F4
      1      0         TOTAL MSGS                    142.0000
      1      0         TOTAL SUCCESSSSFUL MSGS       142.0000
      1      0         AVERAGE MESSAGE DELAY          46.1679
      1      0         TOTAL MISSION THREADS           5.0000
      1      0         TOTAL SUCCESSFUL THREADS        5.0000
      1      0         AVERAGE THREAD DELAY          811.4762
*
*  PRECEDENCE 5 STATISTICS (HIGHEST PRECEDENCE)
*
      1      0         TOTAL MESSAGES                 20.0000
      1      0         SUCCESSFUL MESSAGES            20.0000
      1      0         AVERAGE DELAY                  25.6854
*
*  PRECEDENCE 4 STATISTICS
*
      1      0         TOTAL MESSAGES                   .0000
      1      0         SUCCESSFUL MESSAGES              .0000
      1      0         AVERAGE DELAY                    .0000
*
*  PRECEDENCE 3 STATISTICS
*
      1      0         TOTAL MESSAGES                 30.0000
      1      0         SUCCESSFUL MESSAGES            30.0000
      1      0         AVERAGE DELAY                  51.2587
*
*  PRECEDENCE 2 STATISTICS
*
      1      0         TOTAL MESSAGES                   .0000
      1      0         SUCCESSFUL MESSAGES              .0000
      1      0         AVERAGE DELAY                    .0000
*
*  PRECEDENCE 1 STATISTICS (LOWEST PRECEDENCE)
*
      1      0         TOTAL MESSAGES                 92.0000
      1      0         SUCCESSFUL MESSAGES            92.0000
      1      0         AVERAGE DELAY                  48.9605
```

Figure 4.2.3 Composite Statistics SFI Output File (COMPOSIT.SFI)

4.1.3 MISSION_THREAD MODEL

The mission thread model is used to represent the sequence of messages that are generated to support a particular mission such as a copperhead fire mission. The first message generated on the mission thread will cause other messages to be generated along the thread until the mission is complete or the thread is broken by lost messages.

To support the concept of mission threads, a command hierarchy needs to be overlayed on top of the unit laydown. This hierarchy indicates a units position within the hierarchy such as Forward Observer (FO), and its relationship to the other units in the hierarchy, such as its affiliation with a given Fire Support Team (FIST). The organization SFI input file is used to specify the command hierarchy. A sample ORG.SFI file is shown in Figure 4.3.1. Each unit in the simulation that will participate as part of a mission thread needs to be specified in the organization hierarchy input file. For each unit, the following information is supplied:

- **Unit ID (NNHHH)** - Identification of the unit specified in the deployment input SFI file.
- **Unit Type (ascii)** - Character field (12 characters) identifying unit type (e.g., "FO" for a forward observer).
- **Adjacent Unit IDs (NNHHH, up to 8)** - Identifiers of units that are adjacent to this node in the command hierarchy. A link between two units only needs to be specified once (i.e., if A lists B as an adjacent unit, B does not have to list A). When sending a message along a mission thread link, if a unit is connected to more than 1 unit with the same destination type, the simulation will use the unit that had previously participated on that mission thread. If none of the units have participated on the mission thread, the simulation will pick the first one it finds.

```
*
*  MILITARY ORGANIZATION HIERARCHY SFI INPUT FILE
*
*  CASESAMPLESEQCODE111111111111222222222222333333333333444444444444555555555555666666666666777777777777888888888888999999999999000000000000
H    110              UNIT ID   UNIT TYPE   ADJ UNIT 1  ADJ UNIT 2  ADJ UNIT 3  ADJ UNIT 4  ADJ UNIT 5  ADJ UNIT 6  ADJ UNIT 7  ADJ UNIT 8
F     S          I          A          I          I          I          I          I          I          I          I
                     1001DIVARTY           2004        0          0          0          0          0          0          0
                     1002BDE FSE           2004        0          0          0          0          0          0          0
                     2004BN FDC            2003     4004       3003          0          0          0          0          0
                     2003BN FSE            4004        0          0          0          0          0          0          0
                     4004FIST              4005     3003          0          0          0          0          0          0
                     4005FO                   0        0          0          0          0          0          0          0
                     5005FO                4004        0          0          0          0          0          0          0
                     3003PLT FDC           3004     3005          0          0          0          0          0          0
                     3004BATTERY 1            0        0          0          0          0          0          0          0
                     3005BATTERY 2            0        0          0          0          0          0          0          0
```

Figure 4.3.1 Organization SFI Input File (ORG.SFI)

To support the input of mission thread information within the simulation, the mission thread SFI input file is used to specify the mission thread information. This information is used to build the mission thread table within the simulation. The simulation supports up to 100 different mission threads and a total of 10,000 mission thread links. Figure 4.3.2 shows a sample mission thread SFI input file. The records in the file contain the following information:

- **Mission Thread ID** - Identification of the mission thread.
- **Source Unit Type (ascii)** - Type of unit (12 characters) receiving the message.
- **Message In (ascii)** - Type of message (12 characters) received. The unit that starts the mission thread does not have a received message. This field is therefore ignored for the unit starting the mission thread.
- **Destination Unit Type (ascii)** - Type of unit (12 characters) to received the new message.
- **Message Out (ascii)** - Type of message (12 characters) to send.
- **Message Size (octets)** - Size of message to send. The maximum message size allowed is based on the maximum segment size specified in the segmentation and reassembly SFI input file. The segmentation and reassembly model does not support messages that exceed 256 segments.
- **Message Precedence** - Precedence of the message to send. Message precedence is between 1 (lowest) and 5 (highest). These precedence levels are mapped into other precedence levels at lower layers. There may not be a one-to-one mapping of these precedence levels to lower layer precedence levels.
- **Processing Delay (seconds)** - Amount of time to process outbound messages.

```
*
*  MISSION THREAD INPUT FILE
*
*  CASESAMPLESEQCODE111111111111222222222222333333333333444444444444555555555555666666666666777777777777888888888888
H     1 8             THREAD ID    SOURCE          MSG IN      DESTINATION   MSG OUT     MSG SIZE   PRECEDENCE      DELAY
F      S                I            A               A           A             A             I   OCTETS    I   (1-5)    F3  SECONDS
*
*  COPPERHEAD FIRE MISSION
                                   1FO                          FIST          CFF                    50          1      10.000    ***
                                   1FIST         CFF            BN FSE        CFF                    50          1      10.000    ***
                                   1BN FSE       CFF            BN FDC        CFF                    50          1      10.000    ***
                                   1BN FDC       CFF            PLT FDC       FIREORD                50          1      10.000    ***
                                   1BN FDC       CFF            FIST          MTO                    50          1      10.000    ***
                                   1FIST         MTO            FO            MTO                    50          1      10.000    ***
                                   1PLT FDC      FIREORD        BATTERY 1     FCOM                   38          1      10.000    ***
                                   1PLT FDC      FIREORD        BATTERY 2     FCOM                   38          1      10.000    ***
                                   1BATTERY 1    FCOM           PLT FDC       SHOT #1                25          1      10.000    ***
                                   1PLT FDC      SHOT #1        FIST          SHOT #1                25          1      10.000    ***
                                   1FIST         SHOT #1        FO            SHOT #1                25          1      10.000    ***
                                   1PLT FDC      SHOT #1        FIST          DESIGN #1              25          1      10.000    ***
                                   1FIST         DESIGN #1      FO            DESIGN #1              25          1      10.000    ***
                                   1BATTERY 2    FCOM           PLT FDC       SHOT #2                25          1      10.000    ***
                                   1PLT FDC      SHOT #2        FIST          SHOT #2                25          1      10.000    ***
                                   1FIST         SHOT #2        FO            SHOT #2                25          1      10.000    ***
                                   1PLT FDC      SHOT #2        FIST          DESIGN #2              25          1      10.000    ***
                                   1FIST         DESIGN #2      FO            DESIGN #2              25          1      10.000    ***
                                   1FO           DESIGN #2      FIST          EOM                    25          3      10.000    ***
                                   1FIST         EOM            PLT FDC       EOM                    25          3      10.000    ***
                                   1PLT FDC      EOM            BATTERY 1     EOM                    25          3      10.000    ***
                                   1PLT FDC      EOM            BATTERY 2     EOM                    25          3      10.000    ***
                                   1PLT FDC      EOM            BN FDC        MFR                    25          5      10.000    ***
                                   1BN FDC       MFR            BN FSE        MFR                    50          5      10.000    ***
                                   1BN FDC       MFR            BDE FSE       MFR                    50          5      10.000    ***
                                   1BN FDC       MFR            DIVARTY       MFR                    50          5      10.000    ***
```

Figure 4.3.2 Mission Thread SFI Input File (THREAD.SFI)

4.1.4 DATA_TERMINAL MODEL

The DATA_TERMINAL model represents the host that is generating and receiving messages to be transmitted and received over the underlying network. The SCENARIO_CONTROL model initiates the transmission of a message by the data terminal as a result of reading the request from the message script or by generating it from a needline description. Messages can also be received from the mission thread model. The data terminal takes the transmission request and generates a request for the MIL-STD-188-220 upper layer model. This provides the transmission request to the top of the protocol stack. Each message passed to the upper layer model is assigned a unique internal message number to be used in tracking the message within the simulation.

If the data terminal receives a message from the scenario control indicating that the message is the beginning of a mission thread, the message is passed to the mission thread model for processing.

After passing the information to the upper layer model, the data terminal model provides the message information to the PERFORMANCE_MEASURES model. The PERFORMANCE_MEASURES model uses this information to start tracking the message. This is when the start time of the message for determining message delay is recorded.

At the destination side, the data terminal receives the messages from the upper layer model. It then notifies the performance measures model that the message has been received. This is when the receive time is recorded. The message delay is computed as the difference between the receive and start times. The message delay will include the overhead from every protocol layer that the message goes through.

If a received message is part of a mission thread, the message is passed to the mission thread model for processing.

4.1.5 RADIO MODELS

4.1.5.1 SINCGARS MODEL

The SINCGARS model allows for specifying ICOM or SIP operations. It provides a means to deploy a community of SINCGARS units that are used to transmit and receive messages between the data terminals. The model accounts for the simultaneous generation of messages needed to be sent by various terminals attached to the radios. For data rates less than 16 kbps, the model determines the effective message Bit Error Ratio (BER) from the environment BER.

The model accounts for frequency synchronization, COMSEC, and bit synchronization through the input of delay and probability values. Transmitter attack and receiver release delays are also accounted for.

The model reads a file containing the BER values of the links between the radios. To simplify this process, if the first record contains a source host id of 0, this BER is used as the default BER for all links where the BER is not explicitly given. If this record is not provided the default BER will be 0.

The ICOM receiver uses majority logic coding to determine the received data. The radio supports the following rates with the given majority logic coding:

- 16 Kbps, 1 out of 1
- 4.8 Kbps, 2 out of 3
- 2.4 Kbps, 3 out of 5
- 1.2 Kbps (or less), 7 out of 13

Part of the SIP changes to SINCGARS is viewing the protocol layers from a different perspective. Although the host interfaces to the INC at the physical layer, the model interfaces to the INC at the IP layer. Looking at the protocol stack shown in Table 4.11.1, the layers above the IP layer are considered as part of the host and the layers from IP on down belong to the SINCGARS INC. The lack of delay in interfacing the host to the INC should be insignificant. Most of the delay will be encountered in the intranet interface of the INC.

The SIP radio replaces the majority logic coding done in the ICOM with a concatenated code. The outer code uses Reed-Solomon coding to protect against burst errors. The inner code uses Nadler coding to protect against random bit errors. The extent of coding used is based on the selected data rate. Table 4.11.2 shows the Reed Solomon coding parameters used for different data rates. The Nadler code is a (12,5) block code used for data rates at 4800 bps or less. In addition to Reed-Solomon and Nadler coding, the SIP radio includes side information to enhance the FEC process. The effect of the SIP coding is performed by using a lookup table.

This table provides the BER with coding given the data rate and the environment BER without coding at 16 Kbps.

The SINCGARS SIP was designed to support both voice and data. The radio breaks channel access opportunities into 18 millisecond slots. The slot size used by the simulation is input from the SINCGARS parameter file. 0 to 5 slots are given to voice and 5 to 32 slots are allocated to data. Slots allocated for data are supported by specifying the offset and limit specifications for each priority. Voice slots are supported by having the highest priority offset set to a non-zero value. The parameter file specifies the intertransmission delay time used to specify the beginning of the network contention period (i.e., when slots generation starts).

The SINCGARS model makes the following assumptions:

- All radios have the same propagation delay. This value is read in from the SINCGARS parameter file.
- The PCOM values provided to the simulation account for the effects due to mutual interference.
- The environment is considered time-stationary, i.e., the links between the radios have a time-wise stationary noise environment. This implies that BER values from the radio do not change with time.

Table 4.11.1. GSS Model to ISO Layer Mapping

LAYER	DESCRIPTION	GSS MODEL	LOCATION
7	APPLICATION	MIL_220_UPPER_LAYER	HOST
6	PRESENTATION		
5	SESSION		
4B	TRANSPORT	SEGMENT_AND_REASSEMBLE	
4A	TRANSPORT	UDP	
3B	INTERNET	IP	SINCGARS INC
3A	INTRANET	MIL_220_INTRANET	
2	DATA LINK	MIL_220_DATA_LINK_LAYER	
1	PHYSICAL	SINCGARS	SINCGARS RADIO

Table 4.11.2. Reed-Solomon Code Parameters for Various Data Rates

Data Rate (bps)	Reed-Solomon Code
9600	(32,24)
4800	(32,24)
2400	(32,12)
1200	(32,6)

SINCGARS_SIP Model Parameters

Figure 4.11.1 shows a sample SINCGARS parameter file. The file contains parameters for both the ICOM and SIP radios. The SINCGARS parameters are obtained from the SINCGARS parameter file at the beginning of the simulation. Parameters needed for both the ICOM and SIP are as follows:

- **Data Rate (Kbps)** - The SINCGARS SIP radio operates at 16 Kbps but supports lower rates by using coding techniques to expand to 16Kbps.
- **Transmitter Attack Delay (milliseconds)** - The time interval from keying on a transmitter until the RF signal amplitude has increased to 90% of its steady state value.
- **Receiver Release Delay (milliseconds)**- The time delay from the end of RF energy at the receiver input that is the result of a normal message completion, until the receiver output is squelched or digital data clock ends.
- **Synchronization Delay (milliseconds)**- The time delay from the start of RF energy at the receiver until the receiver is synchronized to the input signal
- **SINCGARS Radio Delays and Synchronization Probabilities (units are for delays)** - Although the delays and synchronization probabilities are separated the simulation simply adds the delays and multiplies the probabilities at the beginning of the simulation.
    - **Frequency Synchronization (milliseconds)** - Overhead associated with frequency synchronization.
    - **COMSEC Synchronization (milliseconds)** - Overhead associated with COMSEC synchronization.
    - **1/0 Preamble (milliseconds)** - Overhead associated with 1/0 Preamble.
    - **Frame Synchronization (milliseconds)** - Overhead associated with frame synchronization.

Parameters that are specific to the SIP are as follows:

- **Intertransmission Delay (milliseconds)** - Amount of time the SIP radio waits before starting the net access period.
- **Radio Access Slot Size (milliseconds)** - Time allocated to an access slot.
- **Radio Access Slot Table** - Table of slot offsets and limits for each priority
  - **Urgent Offset** - Data slot offset for urgent precedence.
  - **Urgent Limit** - Data slot limit for urgent precedence.
  - **Priority Offset** - Data slot offset for priority precedence.
  - **Priority Limit** - Data slot limit for priority precedence.
  - **Routine Offset** - Data slot offset for routine precedence.
  - **Routine Limit** - Data slot limit for routine precedence.

```
*
*  SINCGARS INPUT SFI FILE
*
*  CASESAMPLESEQCODE11111111111222222222222333333333333
H     1 3S              SINCGARS PARAMETERS     VALUE
F                    A            A            F5
*
*  SINCGARS ICOM PARAMETERS
*
                     DATA_RATE                  16000.00000  BITS/SEC
                     TRANSMITTER_ATTACK_DELAY      20.00000  MILLISECONDS
                     RECEIVER_RELEASE_DELAY         0.00000  MILLISECONDS
                     SYNCHRONIZATION_DELAY         20.00000  MILLISECONDS
                     FREQ_SYNCH_DELAY               0.00000  MILLISECONDS
                     COMSEC_SYNCH_DELAY            50.00000  MILLISECONDS
                     PREAMBLE_DELAY                10.00000  MILLISECONDS
                     DRA_FRAME_SYNCH_DELAY          0.00000  MILLISECONDS
                     FREQ_SYNCH_PROB                1.00000
                     COMSEC_SYNCH_PROB              1.00000
                     PREAMBLE_PROB                  1.00000
                     DRA_FRAME_SYNCH_PROB           1.00000
*
*  SINCGARS SIP PARAMETERS
*
                     DATA_RATE                  16000.00000  BITS/SEC
                     TRANSMITTER_ATTACK_DELAY      20.00000  MILLISECONDS
                     RECEIVER_RELEASE_DELAY         0.00000  MILLISECONDS
                     SYNCHRONIZATION_DELAY          5.00000  MILLISECONDS
                     FREQ_SYNCH_DELAY               0.00000  MILLISECONDS
                     COMSEC_SYNCH_DELAY            50.00000  MILLISECONDS
                     PREAMBLE_DELAY                10.00000  MILLISECONDS
                     DRA_FRAME_SYNCH_DELAY          0.00000  MILLISECONDS
                     FREQ_SYNCH_PROB                1.00000
                     COMSEC_SYNCH_PROB              1.00000
                     PREAMBLE_PROB                  1.00000
                     DRA_FRAME_SYNCH_PROB           1.00000
                     INTERTRANSMISSION DELAY       18.00000  MILLISECONDS
                     RADIO ACCESS SLOT SIZE        18.00000  MILLISECONDS
*                    RADIO ACCESS SLOT TABLE
                       URGENT OFFSET                0.00000  SLOT
                       URGENT LIMIT                 3.00000  SLOT
                       PRIORITY OFFSET              0.00000  SLOT
                       PRIORITY LIMIT               3.00000  SLOT
                       ROUTINE OFFSET               0.00000  SLOT
                       ROUTINE LIMIT                3.00000  SLOT
```

Figure 4.11.1 SINCGARS Parameter SFI Input File (SINCGARS.SFI)

4.1.5.2 HAVE_QUICK MODEL

The Have Quick model is used to represent the operation of the Have Quick II radio. It provides a means to deploy a community of Have Quick units that are used to transmit and receive messages between the data terminals. The model accounts for the simultaneous generation of messages needed to be sent by various terminals attached to the radios.

Unlike the SINCGARS model, the Have Quick model is dependent on the frequency hopping algorithm. With the SINCGARS radio, the effects of the frequency hopping were transparent since the SINCGARS radio provides a digital interface and buffers data while changing frequencies. Have Quick does not provide a digital interface and effectively losses data when changing frequencies. This is not a significant problem when used for voice since voice contains enough redundancy to survive the occasional loss of data. Transmitted data, however, is heavily effected by these outages.

Although the simulation will be developed and tested in an unclassified environment, it is possible that running the simulation with the actual Have Quick hopping parameters will require the use of a classified facility. To avoid the need for any classified input files, the following parameters are prompted for at the terminal when running the simulation:

- Average Hop Rate - The Have Quick radio dwell times are randomly determined with the use of a random number sequence generated within the radio. These dwell times are a function of the average hop rate. As with the actual radio, the model will take as input the average hop rate. This rate is specified in hops/second.

- Code Generation Rate - The code generation rate is the rate that the radio generates a new code word to determine the next dwell time and frequency to transmit with. This rate is specified in codes/second.

The Have Quick model assumes that COMSEC synchronization can be achieved in the minimum statistically long dwell time when frequency hopping is being used.

HAVE_QUICK_LINK Model Parameters

Figure 4.12.1 shows a sample Have Quick parameter file. The Have Quick parameters are obtained from the Have Quick parameter file at the beginning of the simulation. These parameters are as follows:

- **Data Rate (Kbps)** - Radio data rate.

- **Transmitter Attack Delay (milliseconds)** - The time interval from keying on a transmitter until the RF signal amplitude has increased to 90% of its steady state value.

- **Receiver Release Delay (milliseconds)**- The time delay from the end of RF energy at the receiver input that is the result of a normal message completion, until the receiver output is squelched or digital data clock ends.

- **Synchronization Delay (milliseconds)**- The time delay from the start of RF energy at the receiver until the receiver is synchronized to the input signal

- **Have Quick Radio Delays and Synchronization Probabilities (units are for delays)** - Although the delays and synchronization probabilities are separated, the simulation simply adds the delays and multiplies the probabilities at the beginning of the simulation.

- **Frequency Synchronization (milliseconds)** - Overhead associated with frequency synchronization.

- **COMSEC Synchronization (milliseconds)** - Overhead associated with COMSEC synchronization.

- **1/0 Preamble (milliseconds)** - Overhead associated with 1/0 Preamble.

- **Other Synchronization (milliseconds)** - Overhead associated with any other synchronization.

- **Hop Recovery Time (milliseconds)** - Amount of time that the radio is unavailable to receive due to frequency changing. This outage time should be used to account for the affects of any skew in the timing clocks.

```
*
*  HAVE QUICK INPUT SFI FILE
*
*  CASESAMPLESEQCODE111111111111222222222222333333333333
H     1 3S            HAVE QUICK PARAMETER      VALUE
F                    A            A           F5
                     DATA RATE                16000.00000  BITS/SEC
                     TRANSMITTER ATTACK DELAY    20.00000  MILLISECONDS
                     RECEIVER RELEASE DELAY       0.00000  MILLISECONDS
                     SYNCHRONIZATION DELAY       20.00000  MILLISECONDS
                     FREQ SYNCH DELAY             0.00000  MILLISECONDS
                     COMSEC SYNCH DELAY          50.00000  MILLISECONDS
                     PREAMBLE DELAY              10.00000  MILLISECONDS
                     OTHER SYNCH DELAYS           0.00000  MILLISECONDS
                     FREQ SYNCH PROB              1.00000
                     COMSEC SYNCH PROB            1.00000
                     PREAMBLE PROB                1.00000
                     OTHER SYNCH PROB             1.00000
                     HOP RECOVERY TIME           15.62500  MILLISECONDS
```

Figure 4.12.1 Have Quick Parameter SFI Input File (HQ.SFI)

4.1.6 GRAPHICS_UTILITY MODEL (other utilities described elsewhere)

The graphics utility model is used to display the deployment of units and the status of the links between them. The input file to the model includes the locations (in screen coordinates) of each of the radios to be deployed. Figure 4.13.1 shows a sample GRAPHICS_UTILITY file. The locations are obtained from the file at the beginning of the simulation. The color of the radio is determined by the network that the radio belongs to (i.e., each network has its own color). The lines connecting the radios indicate the radio connectivity as derived from the connectivity table within the simulation. The style of the line used for each BER threshold is specified in the BER line table input file. These style values indicate which RTG line style to use. Figure 4.13.1 shows a sample BER line table input file. This provides a visual feedback of the connectivity being used by the simulation.

The links between the radios are color coded to indicate the activity on the links. The color coding is a follows:

- **White** - Link is idle (i.e., neither radio is transmitting).
- **Green** - Link is busy (i.e., there is an active transmission going on between the radios).
- **Red** - Collision on the link (transmission was unsuccessful due to a collision).

In addition to the connectivity information, an instrument indicating the percent of collisions in the simulation is displayed and continuously updated during the simulation.

```
*
*  BER LINE TABLE INPUT SFI FILE
*
*  CASESAMPLESEQCODE1111111111222222222222
H     1 2S             BER THRESH  LINE STYLE
F                    E           I
                            0E-00         1  25   FULL LINE
                            1E-05         2  23
                            1E-04       4    17
                            1E-03       9    10
                            1E-02      12     2
                            1E-01      10     1
                            2E-01             0   NO LINE
```

Figure 4.13.1 BER Line Table SFI Input File (BER_LINE.SFI)

4.2 TCP_STACK MODEL

NOTE: Only the TCP model has not been tested. The UDP has not.

4.2.1 COUL MODEL

To provide an upper layer that uses TCP to transmit messages, a Connection Oriented Upper Layer (COUL) model was developed. This model interfaces the data controller to the TCP layer. The model provides the necessary functions to convey messages between hosts without the excess overhead associated with protocols such as FTP.

The COUL model maintains connections between source and destination pairs. When a source has a message to transmit, it first determines whether a connection exists to the destination. If more than 1 destination is indicated, the message is copied with each copy of the message having a single destination and treated as separate messages. If a connection exists, the message is sent over that connection.

If a connection does not exist to a message destination, the COUL model will request a connection from the TCP layer. Once the connection has been established, the message will be sent over the TCP connection.

As with most protocols using TCP, the COUL model makes use of the client/server model. It is assumed that each unit makes a passive open with TCP on the server side on a well known TCP port. This passive open allows the unit to accept a connection from any other unit acting as a client. When another unit desires to set up a connection, it performs an active open with TCP to the server at the destination. When performing the active open, TCP will provide the client with an available port on that machine.

Since TCP connections are bi-directional, once the connection is established, both the client and server can use the connection to send messages. With this connection establishment procedure, it is possible that 2 units can make simultaneous connections to each other thus resulting in 2 connections between these units. Providing special mechanisms to reduce these 2 connections into a single connection would cause more overhead than simply allowing the 2 connections to exists. This case should also be rare and not occur often. Therefore, it has been decided to allow these connections to exist if they are set up.

Since TCP provides a reliable connection with end-to-end acknowledgment, the COUL model will not implement any acknowledgment features. Any failures to transmit a message will be indicated by the TCP protocol.

The COUL model makes use of an idle timer to indicate connections that are not being used. Every time that a message is sent or received on the client side of the connection, the idle timer is restarted. If the idle time expires, the connection is closed. The length of the idle time is user specified. This timer allows the COUL model to be run in various modes. Setting the timeout to a very high value will cause the connection to stay up once it has been established. Setting it to a very low value will cause it to open a connection for every message to be sent.

COUL Model Parameters

Figure 4.5.1 shows a sample connection oriented upper layer parameter file. The parameters are obtained from the COUL parameter file at the beginning of the simulation. These parameters are as follows:

- **Header Size (octets)** - COUL protocol header size.
- **Processing delay (milliseconds)** - COUL processing delay.
- **Idle Timeout (seconds)** - Idle time before connection is closed.

```
*
*  COUL SFI FILE
*
*  CASESAMPLESEQCODE111111111111222222222222333333333333
H     1 3S              COUL PARAMETERS         VALUE
F                 A           A           F5
                  HEADER SIZE                 10.00000  OCTETS
                  PROCESSING DELAY             1.00000  MILLISECONDS
                  IDLE TIMEOUT              1200.00000  SECONDS
```

Figure 4.5.1 COUL Parameter SFI Input File (COUL.SFI)

4.2.2 TCP MODEL

The Transmission Control Protocol (TCP) model supports a reliable connection-oriented transport protocol within the SAT_COMM simulation. It is used to provide a reliable, sequenced, communication path for the upper layer.

The TCP model provides the following capabilities for the upper layer:

- Opening a connection to a remote host.
- Transmitting and receiving data.
- Closing the connection.

TCP uses a 3-way handshake to establish connections. This method insures that both directions of the connection are set up and supports simultaneous connection initiation from both sides. As each side of the connection is set up, each unit queries the system for its network message transmission unit (MTU). This MTU is used to compute the connections receive message segment size (MSS). The sending MSS is received from the other unit as an option to the TCP header during connection establishment. The initial effective MSS used for the connection is the minimum between the sending and receiving MSS values.

TCP does not support the specification of precedence when sending out information over a connection. To maintain the use of precedence within the simulation, the TCP model was slightly modified to support minimal precedence support. The TCP connection establishment messages are sent with the priority of the message that causes the connection to be set up. Each connection maintains a precedence field to store the precedence of the connection. This field is updated every time a new message is sent over the connection.

Connection synchronization is maintained through the use of send and acknowledgment sequence numbers. Once the connection is established, TCP tracks the information to be sent as a continuous data stream. This stream of data is broken into segments and transmitted to the destination. The size of the segments are based on the effective MSS of the connection. At any time during the connection, information can be received from the IP model indicating that the path MTU has changed and a new effective MSS needs to be computed.

The TCP model supports the use of delayed acknowledgments. When a segment arrives that needs to be acknowledged, the delayed acknowledgment timer is started. When the timer expires, the acknowledgment is sent. The delayed acknowledgment timer also acknowledges other segments that arrive before the acknowledgment timer expires.

Segments that are not timely acknowledged are retransmitted when the retransmission timer has expired. The delay used by the retransmission timer is dynamically computed based on the smoothed round trip delay of the connection. The smoothed round trip delay used is a weighted average of individual round trip values for each segment transmitted. As each segment is transmitted, the time it is sent is stored in the retransmission buffer. When an acknowledgment arrives, the difference in the current time and the transmitted time is the round trip time for the connection. In addition to computing the smoothed round trip delay, the round trip deviation from the existing smoothed round trip is computed. A time waited average of the deviation is added to the smoothed round trip to account for variances in the network. The final timeout value used by the retransmission timer is the smoothed round trip time with deviation multiplied by a user specified safety factor that is greater than 1.

If a segment is not acknowledged by the time the retransmission timer expires, the connection enters the retransmission mode. Once in retransmission mode, segments are resent until they are acknowledged or the maximum number of retransmissions has been reached. For each retransmission of a segment, the retransmission timeout delay is doubled. This is to allow the segment acknowledgment more time to be received with each retransmission. The retransmission timeout delay is, however, bounded by a maximum value. If the packet cannot be acknowledged within the maximum number of retransmissions, TCP informs the upper layer that the connection has failed. TCP then aborts the connection.

Acknowledgments that are received while the connection is in the retransmission mode are not used to determine the smoothed round trip delay. This is because it is unknown which transmission of the segment is being acknowledged. Once all outstanding segments have been acknowledged, the connection reverts back to the transmission mode and continues as normal.

TCP assumes that lost segments are due to congestion within the network. To avoid congestion, a congestion window is used. Initially, a connection starts with a congestion window equal to the maximum segment size. Each time a segment is correctly acknowledged, the congestion window is increased by another maximum segment size unless it exceeds the maximum congestion window size. If the connection enters the retransmission state, the maximum congestion window size is lowered to the current congestion window size and the congestion window is reset to the maximum segment size.

TCP creates a Transmission Control Block (TCB) for each connection that it maintains. The TCB contains information needed to support communications between the local and remote TCP ports. This information includes send and receive variables used to sequence packets and support the windowing protocol.

To avoid having old packets within the network effecting newly created connections, TCP waits for a minimum amount of time before allowing a port number to be reused at a host that initiates the closing of the connection. This timeout period is referred to as the TCP quiet time and is set to 2 times the Maximum Segment Lifetime (MSL).

When the upper layer gets a connection from TCP, it is given a socket number. This socket number is the index of a newly created TCB within the TCB table. All further communication between the upper layer and TCP includes the socket number to identify the connection being referenced.

The TCP indication data provided back to the upper layer includes the following:

- Open confirmation and indication
- Received data
- Connection Termination
- Error Status

TCP relies on the use of the IP layer to transmit segments between the source and destination TCP layers. The TCP header used contains the following information:

- Source Port (2 octets)
- Destination Port (2 octets)
- Sequence Number (4 octets)
- Acknowledgment Number (4 octets)
- Data Offset (4 bits)
- Flags (6 bits)
  - Urgent Flag (1 bit)
  - Acknowledgment Flag (1 bit)
  - Push Flag (1 bit)
  - Reset Flag (1 bit)
  - Synchronization Flag (1 bit)
  - Finish Flag (1 bit)
- Window (2 octets)
- Checksum (2 octets)
- Urgent Pointer (2 octets)

Only active opens are modeled. Passive open commands by the receiver are assumed to have been make. It is assumed that there is a well defined port used by the upper layer protocol. It is also assumed that the upper layer has performed a read request when the connection is opened. Received information will be provided to the upper layer when it has arrived without having to receive an explicit read request.

Although TCP provides a checksum to insure that no errors are propagated to upper layers, the checksum is not modeled since any bit errors received will have been detected at the lower layers.

The TCP model makes the following assumptions:

1. The push flag is always used (i.e., TCP will send information immediately without waiting for more information to send).

2. Urgent data as defined by TCP is not used.

3. Source and destination have the appropriate precedence and security settings to set up connections.

4. Explicit passive open and read requests by the upper layer are not modeled.

5. ULP timer is not invoked by the upper layer. The upper layer waits until the maximum number of retransmissions has been exceeded.

TCP Model Parameters

Figure 4.6.1 shows a sample TCP parameter file. The TCP parameters are obtained from the TCP parameter file at the beginning of the simulation. These parameters are as follows:

- **Processing delay (milliseconds)** - TCP processing delay.

- **Maximum Segment Size (octets)** - Maximum size of segments to be generated.

- **Maximum number of retransmissions** - Number of times to retransmit a segment before closing a connection and notifying the upper layer.

- **Maximum Segment Lifetime (seconds)** - maximum assumed duration of a segment within a network.

- **Alpha** - Used to determine the smoothed round trip time for the retransmission timeout. The RFC indicates using values between 0.8 and 0.9.

- **Beta** - Delay variance factor used to determine the retransmission timeout. The RFC indicates using values between 1.3 and 2.0.

- **Lbound (seconds)** - Lower bound of the retransmission timer.

- **Ubound (seconds)** - Upper bound of the retransmission timer.
- **Maximum Receive Buffer (octets)** - Size of buffer available to TCP at the receiver.
- **Acknowledgment Delay (seconds)** - Delay experienced before generating an acknowledgment. May be set to 0 for no delay.

```
*
*  TCP SFI FILE
*
*  CASESAMPLESEQCODE1111111111112222222222222333333333333
H     1 3S                TCP PARAMETERS          VALUE
F                   A            A            F2
                    PROCESSING DELAY                  1.00  MILLISECONDS
                    MAXIMUM SEGMENT SIZE            576.00  OCTETS
                    MAXIMUM RETRANSMISSIONS           4.00
                    MAX SEGMENT LIFETIME            600.00  SECONDS
                    ALPHA                             0.80
                    BETA                              2.00
                    LBOUND                            1.00  SECONDS
                    UBOUND                          300.00  SECONDS
                    MAXIMUM RECEIVE BUFFER      1000000.00  OCTETS
                    ACKNOWLEDGMENT DELAY              0.50  SECONDS
```

Figure 4.6.1 TCP Parameter SFI Input File (TCP.SFI)

4.3 MIL_STD_188_220 MODEL

4.3.1 IP MODEL

The current direction of the DoD is to use commercial-off-the-shelf (COTS) products. In the communications environment, this has been loosely translated to using the Internet Protocol (IP). IP is a protocol that provides routing through interconnected networks/subnetworks.

Traditional IP implementations have been based on version 4 of IP. A new version of IP, however, has recently been proposed to alleviate some of the problems that have been experienced by version 4. This new version is IP version 6 (IPv6) and is defined in RCF 1883. The main improvements of IP that are focused on in the IP model include:

- Conforming to the newly defined IP header.
- Providing end-to-end fragmentation.
- Performing path MTU discovery using ICMPv6 messages.

To support internet routing, each host needs an IP address (i.e., network id and host id). The network id is used to perform routing between networks. The host id is the same as the local network host id. For this model the subnetworks are considered to be networks. The format of the IP header is as follows:

- **Version (4 bits)** - The model will initially only support version 6 of IP. The model will be built, however, to eventually support specifying the version on a network basis.
- **Priority (4 bits)** - The priority field is used to specify the type of traffic. The use of this field is not modeled. The model will, however, continue to support passing the precedence of the messages through the network. The precedence levels are:
  - Flash Override (5)
  - Flash (4)
  - Immediate (3)
  - Priority (2)
  - Routine (1)
- **Flow Label (24 bits)** - The use of flows is not supported by the model.
- **Payload Length (16 bits)** - The length of the IP payload in octets.

- **Next Header (8 bits)** - Type of header following the basic IP header. IPv6 has less information in the basic header than IPv4. It provides for additional optional extension headers, however, to make up for this. This field is used to indicate the next header type in the chain. The last "next header" field in the chain indicates the upper layer protocol header to follow. The values used are the same as the IPv4 protocol field.

- **Hop Limit (8 bits)** - The hop limit is decremented as the IP packet traverses the network to avoid misrouted packets from congesting the network indefinitely. Since errors in the protocols are not being implemented, the time to live field is not necessary in the model and is not implemented.

- **Source and Destination Address (128 bits each)** - The source and destination addresses are internet addresses that identify the source and destination host. The destination address may be a directed broadcast address containing the address of the subnet that is to receive the IP packet at the subnet's gateway.

If fragmentation is being performed, the basic IP header is followed by a fragment header. The format of the fragment header is as follows:

- **Next Header (8 bits)** - Type of header following the fragment header.

- **Reserved (8 bits)** - These bits are reserved for future use.

- **Fragment Offset (13 bits)** - The fragment offset is used to support fragmentation and is not explicitly modeled.

- **Reserved (2 bits)** - These bits are reserved for future use.

- **M Flag (1 bit)** - Indicates if this is the last fragment of the datagram. A 1 indicates that there are more fragments to come.

- **Identification (32 bits)** - Each IP packet is assigned a unique identifier to be used in reassembling IP packets that have been fragmented.

Although some of the header fields are not explicitly modeled, they are accounted for in the header size.

To support routing between subnetworks, the IP model supports the use of gateways. Each gateway supports an interface (i.e., radio) on 2 subnets. Each subnet connection has its own IP address. An SFI input file specifies the gateways in the deployment by providing both IP addresses associated with each gateway. This input file is used to determine the connectivity between subnets. A shortest path algorithm is used to determine how routing should be performed between subnets. The routing information is updated every time a new gateway is added to the simulation. The algorithms are based on the number of subnets that have to be traversed. Dynamic decisions based on network delays are not considered in this version of the model. The gateways only support the IP and lower layers (i.e., the gateway cannot be used as a source or destination).

The IP model supports the IP option to handle multiple destinations within a single subnet as defined for use with IPv4. Once the destination subnet has been reached, the IP layer provides the list of destinations with the data to be sent to the intranet layer. If a group consists of a single destination, the multi destination option is not used and the destination address is put directly in the header.

To support the routing, an IP routing utility was created. This utility reads in the gateway information and creates a routing table. Figure 4.7.1 shows a sample gateway input file. This file is a time ordered file and allows gateways to be added at any time during the simulation. Gateways can also be added from the graphics workstation. As gateways are added to the simulation, a time ordered output file is also built consisting of gateways added from the input file and the graphics workstation. This file has the same format as the input file and can be used as a future input file. Figure 4.7.2 shows a sample gateway output file. The routing utility is also called to generate the IP packet destination header information for each packet that needs to be transmitted.

The IP model supports path MTU discovery through the use of ICMPv6 messages. For each active destination a path MTU is kept in the path MTU utility. If the destination is newly active and there is no path MTU, it is initialized to the network MTU. If an ICMPv6 message arrives indicating that the path MTU is too large, the path MTU is reduced to the size indicated in the ICMP message. Once a path MTU has been reduced, IP starts a timer to indicate when the path MTU should be reset to the network MTU. Any time the path MTU is updated, IP notifies the upper layer of the path MTU change.

If the upper layer requests to send a segment that exceeds the path MTU, the IP layer will break the segment into fragments to be individually delivered and reassembled at the destination. The IP reassembly utility will be used to reassemble the fragments at the destination. A timer will be used by the utility to drop partially reassembled messages that are not completed before the reassemble timeout expires. Unlike IPv4, only the source can perform fragmentation.

```
*
*  GATEWAY INPUT SFI FILE
*
*  CASESAMPLESEQCODE11111111111222222222222
H     1 2S            IPADDR 1    IPADDR 2
F                   I          I
                          01005       02005
                          01004       03004
                          02001       04003
                          02002       05002
                          03001       04002
                          04001       05001
```

Figure 4.7.1 Gateway SFI Input File (GATEWAY.SFI)

```
*
* GATEWAY OUTPUT SFI FILE
*
H     1 2             IPADDR 1    IPADDR 2
F                   I          I
            0              1005        2005
            0              1004        3004
            0              2001        4003
            0              2002        5002
            0              3001        4002
            0              4001        5001
```

Figure 4.7.2 Gateway SFI Output File (GATE_OUT.SFI)

IP Model Parameters

Figure 4.7.3 shows a sample IP parameter file. The IP parameters are obtained from the internet parameter file at the beginning of the simulation. These parameters are as follows:

- **IP Layer Processing Delay (milliseconds)** - Delay experienced by information passing through the internet layer.
- **Path MTU Reset Delay (seconds)** - Amount of time to elapse after decreasing the path MTU until it is reset to the unit's network MTU. A user specified delay of 0 will indicate that the path MTU should never be increased once it has been reduced.
- **Reassembly Timeout (seconds)** - Amount of time allocated to receiving all fragments of a segment after the first fragment has arrived.

```
*
*  IP SFI FILE
*
*  CASESAMPLESEQCODE11111111111222222222222333333333333
H     1 3S                IP PARAMETERS          VALUE
F                   A             A          F5
                    PROCESSING DELAY              1.00000  MILLISECONDS
                    PATH MTU RESET DELAY       1200.00000  SECONDS
                    REASSEMBLY TIMEOUT          120.00000  SECONDS
```

Figure 4.7.3 IP Parameter SFI Input File (IP.SFI)

4.3.2 MIL_220_INTRANET MODEL

The MIL_220_INTRANET model supports routing within a single subnet. Routing is accomplished using directed source relay routing. This is the only type of routing currently supported by the standard. The header, however, allows for future additions of other routing algorithms.

The intranet header is 6 octets long (including originator address) plus 2 octets for each destination and relay. It has the following information:

- **Version Number (4 bits)** - The current value of the MIL-STD-188-220A intranet version number is 0. The version number is not explicitly modeled. The model assumes that all units are using the same version number.

- **Message Type (4 bits)** - The message type is not explicitly modeled. All packets sent are assumed to be derived from the layer model using the intranet layer model. Explicit implementation of the network topology packets are not modeled.

- **Intranet Header Length (8 bits)** - The header length is computed and added to the length of the message to be sent. This updated length is used by the data link layer model to determine the size of the PDU to be sent. The intranet header length is not explicitly included in the header information sent by the model. Instead, the simulation includes the number of destination/relays included in the directed source relay path created by the intranet model. The intranet header length is 6 + 2*(destinations or relays) octets long.

- **Type of Service (8 bits)** - The type of service field is used to pass the message precedence (3 bits) and the delay (D), throughput (T), and reliability (R) flags. The DTR flags are not explicitly modeled. The message precedence levels supported include:

    - Flash Override (5)

    - Flash (4)

    - Immediate (3)

    - Priority (2)

    - Routine (1)

- **Message Identification Number (8 bits)** - The message identification number along with the originating address is used to uniquely identify a message within the system. The originator simply increments the message identification number for each message sent. The number is between 0 and 255 inclusive.

- **Maximum Number of Hops (4 bits)** - The max hop count is used to specify the number of times the message can be relayed within the intranet. Each time the message is relayed, the max hop count is decremented. Once it reaches 0 it will not be relayed. For convenience in the model, the hop count is started at 0 and is incremented every time the frame is relayed.

- **Spare (4 bits)** - Reserved for future use.
- **Originator Address (8 bits)** - The originator address is the intranet address of the source unit.
- **Destination Status and Address Information (16 bits each)** - Since every subnet can reuse the individual addresses, it is necessary to add a subnet specifier to the address in the traffic file to distinguish between the host ids across the subnets. To accomplish this, the host id consists of 5 digits. The first 2 digits consist of a subnet identifier and the last 3 digits contain the host id. The format is NNHHH (NN - subnet id, HHH - host id). For examplė, host 5 in subnet 2 would be addressed as 02005. Each destination/relay address in the directed source relay path is proceeded by a status byte. The status byte contains the following information:
    - **Distance (3 bits)** - Distance (hops) of destination/relay from the originator.
    - **Relay Flag (1 bit)** - Indicates whether the following address should act as a relay.
    - **S/D (2 bits)** - Indicates type of routing to use. Only directed source relay routing is currently supported. Since only one type of routing is being modeled, the S/D specification is not explicitly modeled.
    - **Destination Flag (1 bit)** - Indicates whether the following address is a destination.
    - **ACK Flag (1 bit)** - Indicates that the intranet layer should perform end-to-end acknowledgment. Acknowledgments at the intranet layer are not modeled.

Although portions of the header may not be explicitly modeled, the model uses the intranet header size input parameter to account for the size of these fields.

During the simulation, as the intranet model routes information, it uses a routing utility to generate the directed source relay path. The utility uses a routing table to build the directed source relay path. The directed source relay path is built by concatenating the paths to every destination of the IP packet and concatenating them. The shortest paths are provided first. The path is then scanned for redundant relays which are removed.

The intranet model processes received frames by analyzing the source path. If the path identifies the node as being a destination, the frame data is passed up to the IP layer. If it indicates that the node is a relay at a distance that meets the hop count, the frame data is sent to the next nodes along the path.

The intranet routing table is built and updated using topology update and topology update request messages. Topology update messages are used to provide a unit's spanning tree to all other units that are 1 hop away. These messages are transmitted with the global address at the data link layer. The topology update messages are sent whenever a unit's spanning tree changes or a topology update request is received. Topology update messages are sent no more often than the minimum update time specified in the intranet parameter SFI input file. The intranet topology update message contains the following information:

- **Topology Update Length (8 bits)** - Length of the topology update data.
- **Topology Update ID (8 bits)** - Topology updates are sequentially numbered modulo 256.
- **Topology Entries (24 bits each)**- Each entry in the update message contains the following information:
    - **Node Address (8 bits)** - Identifier of the unit being described.
    - **Node Status (8 bits)** - The status of the unit including:
        - **Link Quality (3 bits)** - Quality of the link. Link Quality is not modeled in the simulation.
        - **Hop Length (3 bits)** - Distance of unit from root of the spanning tree. The simulation does not send the hop length in the topology update message. Hop lengths are recomputed when building the spanning tree from the node predecessor address information.
        - **NR (1 bit)** - Indicates whether the unit is acting as a relay or not. The NR bit is not implemented in the simulation. All units in the simulation act as relays.
        - **Quiet (1 bit)** - Indicates whether the unit is in quiet mode. The simulation does not implement quiet mode.
    - **Node Predecessor Address (8 bits)** - Identifier of the unit's parent in the spanning tree.

Topology update request messages are used to request another unit to send its spanning tree. A topology update request message is sent whenever a unit directly hears another unit that is not marked as 1 hop away in its spanning tree. It also sends a topology update request if a topology update ID in a received transmission header does not match the expected topology update ID. Up to 2 request messages can be sent in the minimum update time specified in the intranet parameter SFI input file. There is no information field associated with a topology update request message.

When a new unit is added to the simulation, it sends a topology update message with only itself in the spanning tree to provide the other units with the information that this unit exists. A randomized delay (less than 1 second) is provided to avoid all hosts from accessing the net at identical times within the simulation. After hearing this unit, the other units update their spanning trees and send their own topology update messages. These messages are used by the new unit to generate its own spanning tree.

MIL_220_INTRANET Model Parameters

Figure 4.8.1 shows a sample intranet parameter file. The MIL_220_INTRANET parameters are obtained from the intranet parameter file at the beginning of the simulation. These parameters are as follows:

- **MIL-STD-188-220 Intranet Layer Processing Delay (milliseconds)** - Every frame that is sent to the IP or data link layer experiences a delay. This delay is the intranet processing delay.

- **Minimum Update Period (seconds)** - Minimum amount of time to elapse between topology update messages.

- **Topology Update Precedence** - Precedence used to send topology update messages. The precedence is between 1 (lowest) and 5 (highest). These precedence levels are mapped into other precedence levels at lower layers. There may not be a one-to-one mapping of these precedence levels to lower layer precedence levels.

- **Topology Update Request Precedence** - Precedence used to send topology update request messages. The precedence is between 1 (lowest) and 5 (highest). These precedence levels are mapped into other precedence levels at lower layers. There may not be a one-to-one mapping of these precedence levels to lower layer precedence levels.

```
*
*  INTRANET SFI FILE
*
*  CASESAMPLESEQCODE1111111111112222222222222333333333333
H     1 3S              INTRANET PARAMETERS        VALUE
F                     A            A          F5
                      INTRANET PROC DELAY          1.00000  MILLISECONDS
                      MINIMUM UPDATE PERIOD        5.00000  SECONDS
                      UPDATE PRECEDENCE            5.00000  1 - 5
                      REQUEST PRECEDENCE           5.00000  1 - 5
```

Figure 4.8.1 Intranet Parameter SFI Input File (INTRANET.SFI)

4.3.3 MIL_220_DATA_LINK MODEL

The following subsections describe the implementation of the MIL-STD-188-220A data link layer.

Data Link Layer Framing

The data link layer framing, shown in Figure 4.9.1, consists of the following fields:

- Transmission Header
- Data Link Frame(s)

These fields are described individually in further detail in the following subsections.

Transmission Header	Data Link Frame(s)

Figure 4.9.1. Data Link Framing

Transmission Header

The Transmission Header is 64 bits long. This value is input from the data link input file instead of being hardcoded. If packet mode transmission synchronization is not being used, the transmission header is FEC and TDC coded within a single TDC block along with the TWC at the physical layer. This TDC block is 168 bits long instead of the normal 384 bit TDC block. The transmission header has the following fields:

- **Flag (8 bits each)** - 8 bit flag at the beginning and end of the transmission header.
- **Transmission Information (16 bits)** - The transmission information consists of flags for FEC, TDC, and scramble as well as the topology update ID and the transmission queue subfield. FEC and TDC are input parameters to the simulation and do not change on a message to message basis. The scramble is not modeled since it does not effect simulation performance. The topology update ID contains the 3 least significant digits of the last intranet topology update message that was sent. The use of the transmission queue subfield is described in further detail in the NAD description.
- **Frame Check Sequence (FCS) (32 bits)** - FCS is assumed to catch any bit errors in the transmission header. Errors that are undetected by the FCS check are highly improbable and are not modeled.

Data Link Frame

The format for the data link frame is shown in Figure 4.9.2. It consists of the following fields:

- **Flag (8 bits each)** - The data link frame is started and ended with an 8 bit flag. When data link frames are concatenated, the ending flag of the first data link frame may be used as the starting flag for the second frame.

- **Source Address (8 bits)** - 8 bit subnet address of the source.

- **Destination Address (8 bits per destination)** - 8 bit subnet address of each destination. Up to 16 destinations can be supported.

- **Control (8 or 16 bits)** - The control field is 8 or 16 bits depending on the type of PDU being transmitted.

- **Information** - The information field can hold up to 3345 octets. This size constraint is known as the maximum transmission unit (MTU) size. The MTU is the maximum size of the information field before zero bit insertion, FEC, and TDC are performed.

- **Frame Check Sequence (FCS) (32 bits)** - The frame check sequence field is 32 bits long. It is assumed that any errors that occur are caught by the FCS check. The model does not account for errors that result in a valid FCS check. The probability of having undetected bit errors within the frame with a 32 bit CRC check is so small that it is impractical to run enough simulations to encounter this situation. The undetected bit errors are not explicitly modeled for that reason.

Flag	Source	Destinations	Control	Information	FCS	Flag
8	8	8 each	8 or 16	...	32	8

Sizes are in bits.

Figure 4.9.2. Data Link Frame Format

Precedence Queues

To support the testing of the NAD algorithms, separate precedence queues are implemented at the data link layer. These queues are used to determine which frames can be concatenated and the order they should be transmitted in.

The MIL-STD-188-220A supports 3 levels of precedence at the data link layer. These precedence are as follows:

- **Urgent (0)** - highest precedence
- **Priority (1)** - medium precedence
- **Routine (2)** - lowest precedence

To support queuing the PDUs to be transmitted, a queue is used for each precedence with higher precedence PDUs being transmitted first. The only exception is when concatenation is being performed as described in the following section.

The intranet layer model supports 5 levels of precedence. The data link layer maps the 5 network layer precedences into the 3 used by the data link layer. The mapping is shown in Table 4.9.1.

Table 4.9.1. Intranet to Data Link Layer Precedence Mapping

INTRANET LAYER	DATA LINK LAYER
Flash Override	Urgent
Flash	Urgent
Immediate	Priority
Priority	Priority
Routine	Routine

In addition to ordering transmissions based on precedence, for a given precedence, type 4 sends information in the following order:

- Acknowledgments
- Retransmissions
- New Transmissions

Data Link Layer Concatenation

The MIL-STD-188-220 standard allows concatenation to occur either at the physical layer or the data link layer. The simulation supports concatenation at the data link layer. This allows more than one frame to be transmitted in a single burst. A switch is included as part of the input to determine whether concatenation should be performed or not.

Only PDUs that do not require coupled acknowledgments are concatenated. The specification allows a PDU to be concatenated with the PDU at the head of the queue even if there is another PDU requiring an acknowledgment before it in the queue. Since the simulation runs in a mode specifying whether acknowledgments are required for every message or not, only adjacent messages in the queue are concatenated (i.e., the queue does not contain a mix of frames that do and don't require acknowledgment). Lower precedence PDUs are also allowed to be concatenated with higher precedence PDUs. The concatenated PDUs cannot exceed the maximum PDU size. The standard supports the use of 1 or 2 flags between concatenated PDUs. The data link parameter input file specifies how many flags to use between concatenated frames. The simulation supports up to 5 PDUs to be concatenated within a frame.

Acknowledged versus Non Acknowledged Transmission

Types 1 and 4 both support acknowledged and non acknowledged transmission at the data link layer. For type 1, acknowledgments are performed in the same net access period as the transmission. This is referred to as a coupled acknowledgment. Each destination calculates its own response hold delay (RHD) to determine when it should transmit an acknowledgment back to the source. The RHD is based on the location of the destination's address within the list of destinations. If TDC is being used, the ACK is repeated a second time within the TDC block to provide 2 opportunities of correct reception. Type 4 acknowledgments are performed in a separate net access period. The simulation provides a switch in the input file indicating if coupled acknowledgments are required or not.

Network Access Delays (NAD)

The MIL_220_DATA_LINK_LAYER model supports the following NAD algorithms:

- Random NAD (R-NAD)
- Prioritized NAD (P-NAD)
- Hybrid NAD (H-NAD)
- Deterministic Adaptive Priority NAD (DAP-NAD)
- Radio Embedded NAD (RE-NAD)

Although the simulation supports the selection of various Network Access Delay (NAD) algorithms, only one algorithm can be used in a single simulation run. Since the simulation input needs to account for any of the algorithms being used, not all of the parameters in the input file are used in every simulation. However, the fields must still be present in the input file. For example, if H-NAD is used, there are several input parameters that need to be specified in the input file that are specific to H-NAD. If H-NAD is not used, the parameter values still need to be included in the input file but their values will not be used.

For the NAD algorithms that use station precedence, the station id within the subnet (i.e., the station id with the subnet portion removed) is used for prioritizing subnet access. If a higher precedence message arrives than what the NAD was calculated for, the NAD is recomputed at the higher precedence.

To support the implementation of the DAP-NAD, the transmission queue subfield of the transmission header includes 2 parameters that controls the performance of DAP-NAD. These parameters are:

- **Data Link Precedence (3 bits)** - Precedence of the highest message within the frame. The message priorities are as follows:

  0 - Urgent

  1 - Priority

  2 - Routine

- **First Subscriber Number (FSN) (6 bits)** - Identifier of host that has the next access opportunity.

To implement the RE-NAD requires implementing the MIL-STD-188-220A scheduler. This scheduler is used to determine when to provide information to be transmitted to the SINCGARS radio. In addition, the transmission queue subfield of the transmission header includes 2 parameters that control the performance of RE-NAD. These parameters are:

- **Queue Precedence (3 bits)** - Precedence of the originators highest precedence queue with information to be transmitted.

- **Queue Length (4 bits)** - Number of concatenated frames needed to empty the highest precedence queue.

Since the network topology does not change during the simulation, the topology factor (FTOP) and partition factor (FPART) parameters will be computed for every node at the beginning of the simulation. The use of RE-NAD immediate mode can be controlled with the data link input file.

To avoid wasting simulation time by computing unnecessary NAD values when a unit has no information to send, each unit only computes the NAD values when it has something to transmit. The expiration of the TP timer is always stored in every radio to be used in determining the basis of the NAD values when there is a PDU to transmit.

Type 4

The SINCGARS SIP uses the type 4 service of MIL-STD-188-220A. Type 4 is similar to type 1 with some modifications. The purpose of type 4 is to provide an uncoupled acknowledgment protocol that does not require the connection setup and maintenance overhead that is required for type 2. The Type 4 implementation includes the following:

- **Uncoupled Acknowledgments** - The type 4 acknowledgments are not performed in the same net access period. Every arriving information PDU that requires an acknowledgment will generate an acknowledgment response PDU. Each information PDU sent by the source contains a unique identification number that is included in the ACK. Acknowledgments are not piggybacked onto other outbound PDUs. The acknowledgment can, however, be concatenated with other outbound PDUs. Acknowledgments are transmitted at the same precedence as the original information PDU it is acknowledging. Acknowledgments are sent before other PDUs at the same precedence.

- **Outstanding PDUs** - Since the acknowledgments are uncoupled, type 4 allows more than one PDU to be sent before receiving an ACK. The number of outstanding PDUs allowed at each source is an input parameter between 5 and 20. This limit is a total for all links. This value is an input parameter to the simulation.

- **Acknowledgment Timers** - Type 4 has an acknowledgment timer for each outstanding PDU. As each timer expires, its associated PDU is offered for retransmission. Retransmissions are sent before new PDUs. The number of transmission attempts is an input parameter.

Type 2

Type 2 provides connection oriented data link services. The specific features of type 2 that are modeled include:

- **Connection Establishment** - Type 2 requires that a connection be established before sending information frames. To support this, the protocol has 2 operating modes to indicate whether the connection has been established or not. The simulation initializes all connections as already being active at the start of the simulation. Connections only go through the connection re-establishment process if the link is lost due to a sequence of lost frames.

- **Sequence Numbers and Status Variables -** To support the transmission of information PDUs, sequence numbers and state variables are kept for every connection. These variables allow the protocol to piggyback acknowledgments on outgoing information PDUs. They are initialized at connection establishment time and when the connection is reset.

- **P/F Bit** - The P/F bit is used by the protocol to force the receiver to reply to a command. This allows the sender to gather information about a connection to determine its status.

- **Acknowledgments** - Acknowledgments are piggybacked on outgoing information PDUs. An acknowledgment timer is used to determine when to go in to the recovery state. If a PDU is received and there are no PDUs to be sent out on the needed connection, an acknowledgment wait timer is used to indicate when a dedicated PDU containing the ACK should be sent.

- **Rejects** - Frames received out of sequence cause the generation of a Reject PDU indicating the last correctly received sequence number. The reject process is supported by a reject timer.

- **Outstanding PDUs** - Unlike the window used by type 4, type 2 supports a different window for each link connection. The standard supports window sizes up to 127. This window size is an input parameter to the simulation.

Data Link Layer Parameters

Figure 4.9.3 shows a sample data link layer parameter file. The data link layer parameters are obtained from the data link layer parameter file at the beginning of the simulation. These parameters are as follows:

- **Processing Delay (milliseconds)** - Delay to account for the processing of transmitted and received frames at the data link layer.

- **Zero Bit Insertion Factor** - Multiplier used to increase data size to account for the overhead of the zero bit insertion algorithm. If no zero bit insertion is desired, a 1 should be used for the factor.

For each type of radio, the following parameters are required:

- **Data Link Layer Type** - Data link layer type to use. Intranet topology update and topology update request messages are always sent using Type1. The simulation supports the following types:

  1. Type 1

  2. Type 2

4. Type 4

- **Concatenation Switch** - Switch to indicate whether to perform data link layer concatenation or not. Since concatenation is not allowed with coupled acknowledgments, if the concatenation switch and acknowledgment switch are both set when type 1 is being used, the acknowledgment switch overrides and the concatenation switch will be set to 0.

  0 - Concatenation not used.

  1 - Concatenation allowed.

- **Acknowledgment Switch** - Switch to indicate whether to use acknowledgments or not. Type 1 uses coupled acknowledgments. Type 4 acknowledgments are uncoupled. Intranet topology update and topology update request messages are always sent without acknowledgment.

  0 - Acknowledgment not required.

  1 - Acknowledgment required.

- **Maximum PDU Size (octets)** - Size of the largest PDU (or group of PDUs) that is included within a single frame. The MIL-STD-188-220A standard supports a maximum PDU size in the range from 708 to 3345 octets.

- **Maximum Transmissions** - Number of transmission attempts that are allowed before dropping the frame. If acknowledgments are not requested, only a single transmission is attempted regardless of this setting.

- **Forward Error Correction (FEC)** - Whether Golay FEC is used:

  0 - No

  1 - Yes

- **Time Dispersal Coding (TDC)** - Whether TDC (384 bit blocks) is used:

  0 - No

  1 - Yes

- **Network Access Delay (NAD) Algorithm** - All 5 NAD algorithms are supported. These are as follows:

    1. Random (R-NAD)
    2. Priority (P-NAD)
    3. Hybrid (H-NAD)
    4. Distributed Adaptable Priority (DAP-NAD)
    5. Radio Embedded (RE-NAD)

- **Network Access Control (NAC) Timing Parameters (defined in Appendix C of MIL-STD-188-220A)**

    - **EPRE (milliseconds)** - Equipment preamble time.
    - **PHASING (milliseconds)** - Phasing transmission time.
    - **S (milliseconds)** - Coupled acknowledgment transmission time
    - **ELAG (milliseconds)** - Equipment lag time.
    - **TTURN (milliseconds)** - Transmitter turnaround time.
    - **RTURN (milliseconds)** - Receiver turn around time.
    - **DTEACK (milliseconds)** - Data terminal equipment acknowledgment preparation time.
    - **DTEPROC (milliseconds)** - Data terminal equipment processing time.
    - **TOL (milliseconds)** - Tolerance time.
    - **B (milliseconds)** - Network busy detect time.

The Type 2 parameters are as follows:

- **Type 2 Window Size** - Number of type 2 information PDUs that can be waiting acknowledgment at any given time on a link.
- **Type 2 Acknowledgment Timeout (seconds)** - Amount of time type 2 waits for an acknowledgment on a link before going into the timer recovery state.

- **P Bit Timer Delay (seconds)** - Amount of time type 2 waits for a response to a PDU sent with the P bit set.
- **Reject Timer Delay (seconds)** - Amount of time type 2 waits for a response to a reject command.
- **Acknowledgment Wait Timer Delay (seconds)** - Amount of time type 2 waits for a piggybacked acknowledgment to be sent on a link before sending a dedicated acknowledgment PDU.
- **Type 2 N2** - Number of times to retransmit a control PDU on a link.

The Type 4 parameters are as follows:

- **Type 4 Window Size** - Number of type 4 PDUs that can be waiting acknowledgment at any given time.
- **Type 4 Acknowledgment Timeout (seconds)** - Amount of time a type 4 PDU waits for an acknowledgment before retransmitting the frame.

The H-NAD parameters are as follows:

- **Percent Urgent** - The approximated percent of traffic on the subnet that is at the urgent level.
- **Percent Priority** - The approximated percent of traffic on the subnet that is at the priority level.
- **Percent Routine** - The approximated percent of traffic on the subnet that is at the routine level.
- **Traffic Loading (TL)** - The amount of net traffic expected. Although the simulation accepts any real value, the standard provides the following guidelines:
    - 1.2 (Heavy Loading)
    - 1.0 (Normal Loading)
    - 0.8 (Light Loading)

The RE-NAD parameters are as follows:

- **T** - Topology Coefficient
- **P** - Partition Coefficient
- **D** - Damping Coefficient
- **Immediate Mode Switch** - Switch to indicate if immediate mode should be used for RE-NAD.

    0 - Immediate mode Disabled

    1 - Immediate mode Enabled

```
*
*  DATA LINK SFI FILE
*
*  CASESAMPLESEQCODE111111111111222222222222333333333333
H     1 3S               DATA LINK PARAMETERS      VALUE
F                   A           A           F5
                    LINK LAYER PROC DELAY      128.00000  MILLISECONDS
                    ZERO BIT INSERT FACTOR       1.00000  1-NONE
*
*  RADIO TYPE 1 (ICOM)
*
                    DLL TYPE                     1.00000  1, 2, OR 4
                    DATA LINK CONCATENATION      0.00000  1-YES, 0-NO
                    ACKNOWLEDGMENTS              1.00000  1-YES, 0-NO
*                     CONCATENATION AND ACKNOWLEDGMENTS CANNOT
*                     BOTH BE SET TO YES FOR TYPE 1
                    MTU                       3345.00000  OCTETS  (RELATED TO SEGMENT SIZE)
                    MAX TRANSMISSIONS            3.00000
*
                    FORWARD ERROR CORRECTION     1.00000  0-OFF, 1-ON
                    TIME DISPERSAL CODING        1.00000  0-OFF, 1-ON
                    NAD ALGORITHM                1.00000
*                     1 - RANDOM NAD
*                     2 - PRIORITY NAD
*                     3 - HYBRID NAD
*                     4 - DISTRIBUTED ADAPTABLE PRIORITY NAD
*                     5 - RADIO EMBEDDED NAD
*
*                   *** TIMING PARAMETERS
                    EPRE                       380.00000  MILLISECONDS
                    PHASING                    500.00000  MILLISECONDS
                    S                           73.00000  MILLISECONDS
                    ELAG                       110.00000  MILLISECONDS
                    TTURN                        0.00000  MILLISECONDS
                    RTURN                        0.00000  MILLISECONDS
                    DTEACK                     390.00000  MILLISECONDS
                    DTEPROC                      0.00000  MILLISECONDS
                    TOL                        500.00000  MILLISECONDS
                    B                           26.00000  MILLISECONDS
*
*  RADIO TYPE 2 (SIP)
*
                    DLL TYPE                     4.00000  1, 2, OR 4
                    DATA LINK CONCATENATION      1.00000  1-YES, 0-NO
                    ACKNOWLEDGMENTS              1.00000  1-YES, 0-NO
*                     CONCATENATION AND ACKNOWLEDGMENTS CANNOT
*                     BOTH BE SET TO YES FOR TYPE 1
                    MTU                       3345.00000  OCTETS  (RELATED TO SEGMENT SIZE)
                    MAX TRANSMISSIONS            3.00000
*
                    FORWARD ERROR CORRECTION     0.00000  0-OFF, 1-ON
                    TIME DISPERSAL CODING        0.00000  0-OFF, 1-ON
                    NAD ALGORITHM                5.00000
*                     1 - RANDOM NAD
*                     2 - PRIORITY NAD
*                     3 - HYBRID NAD
*                     4 - DISTRIBUTED ADAPTABLE PRIORITY NAD
*                     5 - RADIO EMBEDDED NAD
*
```

Figure 4.9.3A Data Link Layer Parameter SFI Input File (DLL.SFI)

```
*                   *** TIMING PARAMETERS
                    EPRE                        380.00000  MILLISECONDS
                    PHASING                     500.00000  MILLISECONDS
                    S                            73.00000  MILLISECONDS
                    ELAG                        110.00000  MILLISECONDS
                    TTURN                         0.00000  MILLISECONDS
                    RTURN                         0.00000  MILLISECONDS
                    DTEACK                      390.00000  MILLISECONDS
                    DTEPROC                       0.00000  MILLISECONDS
                    TOL                         500.00000  MILLISECONDS
                    B                            26.00000  MILLISECONDS
*
*  RADIO TYPE 3 (HAVE QUICK)
*
                    DLL TYPE                      4.00000  1, 2, OR 4
                    DATA LINK CONCATENATION       1.00000  1-YES, 0-NO
                    ACKNOWLEDGMENTS               1.00000  1-YES, 0-NO
*                     CONCATENATION AND ACKNOWLEDGMENTS CANNOT
*                     BOTH BE SET TO YES FOR TYPE 1
                    MTU                        3345.00000  OCTETS  (RELATED TO SEGMENT SIZE)
                    MAX TRANSMISSIONS             3.00000
*
                    FORWARD ERROR CORRECTION      1.00000  0-OFF, 1-ON
                    TIME DISPERSAL CODING         1.00000  0-OFF, 1-ON
                    NAD ALGORITHM                 4.00000
*                     1 - RANDOM NAD
*                     2 - PRIORITY NAD
*                     3 - HYBRID NAD
*                     4 - DISTRIBUTED ADAPTABLE PRIORITY NAD
*                     5 - RADIO EMBEDDED NAD
*
*                   *** TIMING PARAMETERS
                    EPRE                       1380.00000  MILLISECONDS
                    PHASING                     500.00000  MILLISECONDS
                    S                            73.00000  MILLISECONDS
                    ELAG                        110.00000  MILLISECONDS
                    TTURN                         0.00000  MILLISECONDS
                    RTURN                      1000.00000  MILLISECONDS
                    DTEACK                      390.00000  MILLISECONDS
                    DTEPROC                       0.00000  MILLISECONDS
                    TOL                         500.00000  MILLISECONDS
                    B                            26.00000  MILLISECONDS
*
*  TYPE 2 PARAMETERS
                    TYPE 2 WINDOW SIZE           64.00000
                    TYPE 2 ACK TIMEOUT          120.00000  SECONDS
                    P BIT TIMEOUT                10.00000  SECONDS
                    REJECT TIMEOUT              240.00000  SECONDS
                    ACK WAIT TIMEOUT             40.00000  SECONDS
                    TYPE 2 N2                     3.00000  RETRANSMISSIONS
*
*  TYPE 4 PARAMETERS
                    TYPE 4 WINDOW SIZE            5.00000
                    TYPE 4 ACK TIMEOUT           30.00000  SECONDS
*
*  H-NAD PARAMETERS
                    PERCENT URGENT               25.00000  ONLY NEEDED FOR H-NAD
                    PERCENT PRIORITY             25.00000  ONLY NEEDED FOR H-NAD
                    PERCENT ROUTINE              50.00000  ONLY NEEDED FOR H-NAD
                    TRAFFIC LOAD                  1.00000  ONLY NEEDED FOR H-NAD
*                     HEAVY  LOADING - 1.2
*                     NORMAL LOADING - 1.0
*                     LIGHT  LOADING - 0.8
*
*  RE-NAD PARAMETERS
                    T - TOPOLOGY COEFF            1.00000  FH-1, SC-2
                    P - PARTITION COEFF           3.00000
                    D - DAMPING COEFF             7.00000
                    IMMEDIATE MODE                1.00000  1-YES, 0-NO FOR RE-NAD
```

Figure 4.9.3B Data Link Layer Parameter SFI Input File (DLL.SFI) - Continued

4.3.4 PHYSICAL_LAYER MODEL

The physical layer includes transmission synchronization overhead added to the information to be sent after the transmission header has been added by the data link layer.

The transmission synchronization supports 3 modes of operation. These modes are as follows:

- Asynchronous Mode
- Synchronous Mode
- Packet Mode

The physical layer parameter input file has a field to specify the transmission mode to be used for each radio type.

Figure 4.10.1 shows the transmission synchronization for the asynchronous and synchronous modes. The bit synchronization field is only required for the asynchronous mode. The fields are as follows:

- **Bit Synchronization (64 or 1680 bits) -** Bit synchronization is only used for asynchronous mode transmission. The 1680 bit synchronization is only used by Have Quick when COMSEC is not being used. There is no explicit bit synchronization check done by the receiver. The model assumes that if the frame synchronization is successful, then the bit synchronization is also successful. The length of the bit synchronization is not hardcoded in the simulation processes but defined once in the physical layer parameter resource.

- **Frame Synchronization (64 bits)** - Frame synchronization can be achieved with up to 13 bit errors. Different frame synchronization parameters are used to indicate if the robust communications protocol is active or not. The frame synchronization field length and maximum allowed bit errors are not hardcoded in the simulation processes but defined once in the physical layer parameter resource.

- **Robust Frame Format (75 bits)** - The robust frame synchronization consists of 7 bits encoded using BCH (15,7) coding and repeated 5 times. The field indicates whether multi-dwell and scrambling are used, the type of multi-dwell format being used, and the convolutional coding constraint length. This field is only used if the robust communications protocol is active.

- **Transmission Word Count (TWC) (12 bits)** - The TWC is FEC and TDC coded within a single TDC block along with the Transmission Header. The TWC is not used to determine if the frame is correctly received or not. This decision is based on the flag boundaries. Only the length of the TWC field is modeled in the simulation. The TWC length is not hardcoded in the simulation processes but defined once in the physical layer parameter resource.

Bit Synch†	Frame Synch	Robust Frame Format††	TWC
64	64	75	12

Sizes are in bits.
† Bit Synch is only used with asynchronous mode.
†† Robust Frame Format only used with robust communications protocol.

Figure 4.10.1. Asynchronous and Synchronous Mode Transmission Synchronization

The packet mode transmission synchronization is different than the asynchronous and synchronous transmission synchronization. Instead of containing a bit synchronization field, a frame synchronization field, and a transaction word count (TWC), the packet synchronization contains at least 4 consecutive 8 bit HDLC flags back to back. The number of HDLC flags used is not hardcoded in the simulation processes but defined once in the physical layer parameter resource.

Robust Communication Protocol

The robust communication protocol is an optional protocol that can be applied at the physical layer of MIL-STD-188-220A. It was developed to overcome the data loss problems of using non-digital radios that do not provide data buffering during frequency hopping such as Have Quick II. It provides robust communications by providing the following features:

- **1/3 Rate Convolutional Coding** - The convolution code is either inactive or operating with a constraint length of 3, 5, or 7. The simulation reads in a table that is used to determine the data BER after decoding given the BER before decoding and the constraint length.

- **Data Scrambling** - Data scrambling does not effect the simulation and is not modeled.

- **Multi-Dwell Protocol (Packetizing)** - Data is packetized before being transmitted. Each packet (except for the first) has a start of packet (SOP) pattern field and a segment counter. Table 4.10.1 shows the various multi-dwell transmission formats that are supported. The segment count field is a BCH (15,7) codeword repeated 1, 3, or 5 times depending on the majority vote option being used. The majority vote option being used depends on the multi-dwell transmission format selected. The number of 64 bit segments in the packet also depends on the multi-dwell transmission format. The multi-dwell protocol determines if data is being dropped by the radio due to a frequency hop and stops transmitting until the hop recovery time has expired. Transmission is resumed at the beginning of the last assumed successfully transmitted segment at the start of a new packet. If duplicate segments are transmitted, only the last is used. Both are not checked to determine which is better.

Table 4.10.1 Multi-Dwell Transmission Format

Format	Majority Vote	Bits in SOP	Segments/Packet
0	1 of 1	32	11
1	2 of 3	64	11
2	3 of 5	64	13
3	3 of 5	64	6

The robust communication protocol specifies that type 1 coupled acknowledgments use multi-dwell transmission format 3. The protocol does not specify which type of multi-dwell transmission format to use for other transmissions. If a message does not exceed a single TDC block, it will be sent using multi-dwell transmission format 3 regardless of format chosen. The simulation will include an input parameter to specify the multi-dwell transmission format to use for transmissions that do not exceed 6 64 bit segments. If a message can be completely transmitted within the minimum statistically long dwell time, it is transmitted without multi-dwell.

Physical Layer Parameters

Figure 4.10.2 shows a sample physical layer parameter file. The data link layer parameters are obtained from the physical layer parameter file at the beginning of the simulation. These parameters are as follows:

- **Processing Delay (milliseconds)** - Delay to account for the processing of transmitted and received frames at the physical layer.

- **1/3 Convolutional Code Constraint Length** - Constraint length of the 1/3 rate convolutional code. The standard supports constraints lengths of 3, 5, and 7. A 0 constraint length is used to indicate that the convolutional code is not used.

- **Multi-Dwell Format** - Multi-dwell format to use. Options are 0 - 3 as defined in Table 4.4.1.

- **Transmission Mode** - The transmission mode used by each radio.

```
*
*  PHYSICAL LAYER SFI FILE
*
*  CASESAMPLESEQCODE11111111111122222222222222333333333333
H     1 3S             PHYSICAL PARAMETERS       VALUE
F                 A           A           F5
                  PHYSICAL LAYR PROC DELAY     1.00000  MILLISECONDS
                  1/3 CODE CONSTRAINT LEN      0.00000  0, 3, 5, OR 7
                  MULTI-DWELL FORMAT           0.00000  0 - 3
*
*  RADIO TYPE 1 (ICOM)
*
                  TRANSMISSION_MODE            2.00000
*                   1 - ASYNCHRONOUS
*                   2 - SYNCHRONOUS
*                   3 - PACKET (USE WITH RE-NAD)
*
*  RADIO TYPE 2 (SIP)
*
                  TRANSMISSION_MODE            3.00000
*                   1 - ASYNCHRONOUS
*                   2 - SYNCHRONOUS
*                   3 - PACKET (USE WITH RE-NAD)
*
*  RADIO TYPE 3 (HAVE QUICK)
*
                  TRANSMISSION_MODE            2.00000
*                   1 - ASYNCHRONOUS
*                   2 - SYNCHRONOUS
*                   3 - PACKET (USE WITH RE-NAD)
```

Figure 4.10.2 Physical Layer Parameter SFI Input File (PHYSICAL.SFI)

5. SUMMARY

This report covers work done on the SBIR Phase I contract, AF Topic No. 99-132, "High Throughput Terminal/CDMA Model for Satellite Communications" for the period November 25 to December 23, 1999. During this period, PSI has completed implementation and test of a Phase I version of the CDMA Satellite simulation and analysis tool. Delivery of this working tool to Rome completes 100% of the proposed Phase I work effort. As part of this delivery, PSI has also provided an upgraded version of GSS Release 8.0, the simulation environment, and scenarios and test demonstrations to the Air Force for use in demonstrations at AFRL, Rome.

PSI hopes that these Phase I deliveries will demonstrate the power of GSS to design, test, and evaluate communication systems. We also hope that we have demonstrated the ability of PSI to quickly provide models and protocols embedded in detailed simulations, at a low cost, to support rapid prototyping for the Air Force.

6. REFERENCES

1. MIL-STD-188-220A, "Interoperability Standard for Digital Message Transfer Device Subsystems," 28 November 1995.

2. MIL-STD-188-220A, "Interoperability Standard for Digital Message Transfer Device Subsystems," 27 July 1995.

3. MIL-STD-188-220 () For Task Force XXI, "Interoperability Standard for Digital Message Transfer Device Subsystems," Final Draft 23 November 1994.

4. MIL-STD-188-220 Draft Military Standard, "INTEROPERABILITY STANDARD FOR DIGITAL MESSAGE TRANSFER DEVICE SUBSYSTEMS," 7 May 1993.

5. MIL-STD-188-220() Protocol Profile for Task Force XXI, Final Draft, 23 November 1994.

6. "SINCGARS System Engineering Document - Ground Radio System," 28 July 1989.

7. RFC 758, "User Datagram Protocol," 28 August 1980.

8. RFC 793. "Transmission Control Protocol," September 1981.

9. RFC 791, "Internet Protocol," September 1981.

10. RFC 919, "Broadcasting Internet Datagrams," October 1984.

11. RFC 922, "Broadcasting Internet Datagrams in the Presence of Subnets," October 1984.

12. RFC 950, "Internet Standard Subnetting Procedure," August 1985.

13. RFC 1112, "Host Extensions for IP multicasting," August 1989.

14. RFC 1122, "Requirements for Internet Hosts -- Communication Layers," October 1989.

15. RFC 1191, "Path MTU Discovery," November 1990

16. RFC 1883, "Internet Protocol, Version 6," December 1985

17. RFC 1885, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification," December 1995

18. "Variable Message Format Technical Interface Design Plan," Volume III.

19. "THE GENERAL SIMLUATION SYSTEM (GSS) USERS MANUAL," Release 6.2, 27 April 1995.

20. "the GENERAL SIMULATION SYSTEM RUN-TIME GRAPHICS (GSS-RTG) USERS REFERENCE MANUAL," Release 4.2.4, 2 June 1995.

21. "Building Reusable Models - Using GSS," 1 March, 1995.

22. Comer, Douglas E. and Stevens, David L., "Internetworking with TCP/IP, Volume II", Prentice Hall, 1991.

23. Huitema, Christian "IPv6 The New Internet Protocol", Prentice Hall, 1996.